July 2024

Search posts from July 2024 Search

120 posts: 4 entries, 81 links, 35 quotes

July 1, 2024

A write-ahead log is not a universal part of durability (via) Phil Eaton uses pseudo code to provide a clear description of how write-ahead logs in transactional database systems work, useful for understanding the tradeoffs they make and the guarantees they can provided.

I particularly liked the pseudo code explanation of group commits, where clients block waiting for their commit to be acknowledged as part of a batch of writes flushed to disk.

# 3:05 pm / databases, phil-eaton

Announcing the Ladybird Browser Initiative (via) Andreas Kling's Ladybird is a really exciting project: a from-scratch implementation of a web browser, initially built as part of the Serenity OS project, which aims to provide a completely independent, open source and fully standards compliant browser.

Last month Andreas forked Ladybird away from Serenity, recognizing that the potential impact of the browser project on its own was greater than as a component of that project. Crucially, Serenity OS avoids *any* outside code - splitting out Ladybird allows Ladybird to add dependencies like libjpeg and ffmpeg. The Ladybird June update video talks through some of the dependencies they've been able to add since making that decision.

The new Ladybird Browser Initiative puts some financial weight behind the project: it's a US 501(c)(3) non-profit initially funded with $1m from GitHub co-founder Chris Chris Wanstrath. The money is going on engineers: Andreas says:

> We are 4 full-time engineers today, and we'll be adding another 3 in the near future

Here's a 2m28s video from Chris introducing the new foundation and talking about why this project is worth supporting.

# 4:08 pm / browsers, ffmpeg, open-source, andreas-kling, ladybird

---

> When presented with a difficult task, I ask myself: *"what if I didn't do this at all?"*. Most of the time, this is a stupid question, and I have to do the thing. But ~5% of the time, I realize that I can completely skip some work.

— **Evan Hahn**

# 8:42 pm / productivity, programming

Russell Keith-Magee: Build a cross-platform app with BeeWare. The session videos from PyCon US 2024 have started showing up on YouTube. So far just for the tutorials, which gave me a chance to catch up on the BeeWare project with this tutorial run by Russell Keith-Magee.

Here are the accompanying slides (PDF), or you can work through the official tutorial in the BeeWare documentation.

The tutorial did a great job of clarifying the difference between Briefcase and Toga, the two key components of the BeeWare ecosystem - each of which can be used independently of the other.

Briefcase solves packaging and installation: it allows a Python project to be packaged as a native application across macOS, Windows, iOS, Android and various flavours of Linux.

Toga is a toolkit for building cross-platform GUI applications in Python. A UI built using Toga will render with native widgets across all of those supported platforms, and experimental new modes also allow Toga apps to run as SPA web applications and as Rich-powered terminal tools (via toga-textual).

Russell is excellent at both designing and presenting tutorial-style workshops, and I made a bunch of mental notes on the structure of this one which I hope to apply to my own in the future.

# 10:49 pm / python, russell-keith-magee, beeware

> I *like* the lies-to-children motif, because it underlies the way we run our society and resonates nicely with Discworld. Like the reason for Unseen being a storehouse of knowledge - you arrive knowing everything and leave realising that you know practically nothing, therefore all the knowledge you had must be stored in the university. *But it's like that in "real Science", too.* You arrive with your sparkling A-levels all agleam, and the first job of the tutors is to reveal that what you thought was true is only true for a given value of "truth".
>
> Most of us need just "enough" knowledge of the sciences, and it's delivered to us in metaphors and analogies that bite us in the bum if we think they're the same as the truth.
>
> — **Terry Pratchett**

# 11:39 pm / analogy, terry-pratchett, science

July 2, 2024

Weeknotes: a livestream, a surprise keynote and progress on Datasette Cloud billing

My first YouTube livestream with Val Town, a keynote at the AI Engineer World’s Fair and some work integrating Stripe with Datasette Cloud. Plus a bunch of upgrades to my blog.

[... 1,124 words]

4:46 am / blogging, datasette, weeknotes, datasette-cloud, stripe

> So VisiCalc came and went, but the software genre it pioneered – the spreadsheet – endured to become arguably the most influential type of code ever written, at least in the sense of touching the lives of millions of office workers. I’ve never worked in an organisation in which spreadsheet software was not at the heart of most accounting, budgeting and planning activities. I’ve even known professionals for whom it’s the only piece of PC software they’ve ever used: one elderly accountant of my acquaintance, for example, used Excel even for his correspondence; he simply widened column A to 80 characters, typed his text in descending cells and hit the “print” key.
>
> — **John Naughton**

# 5:23 am / spreadsheets, excel

Optimizing Large-Scale OpenStreetMap Data with SQLite (via) JT Archie describes his project to take 9GB of compressed OpenStreetMap protobufs data for the whole of the United States and load it into a queryable SQLite database.

OSM tags are key/value pairs. The trick used here for FTS-accelerated tag queries is really neat: build a SQLite FTS table containing the key/value pairs as space concatenated text, then run queries that look like this:

```
SELECT
    id
FROM
    entries e
    JOIN search s ON s.rowid = e.id
WHERE
    -- use FTS index to find subset of possible results
    search MATCH 'amenity cafe'
    -- use the subset to find exact matches
    AND tags->>'amenity' = 'cafe';
```

JT ended up building a custom SQLite Go extension, SQLiteZSTD, to further accelerate things by supporting queries against read-only zstd compresses SQLite files. Apparently zstd has a feature that allows "compressed data to be stored so that subranges of the data can be efficiently decompressed without requiring the entire document to be decompressed", which works well with SQLite's page format.

# 2:33 pm / go, openstreetmap, sqlite, zstd

Compare PDFs. Inspired by this thread on Hacker News about the C++ diff-pdf tool I decided to see what it would take to produce a web-based PDF diff visualization tool using Claude 3.5 Sonnet.

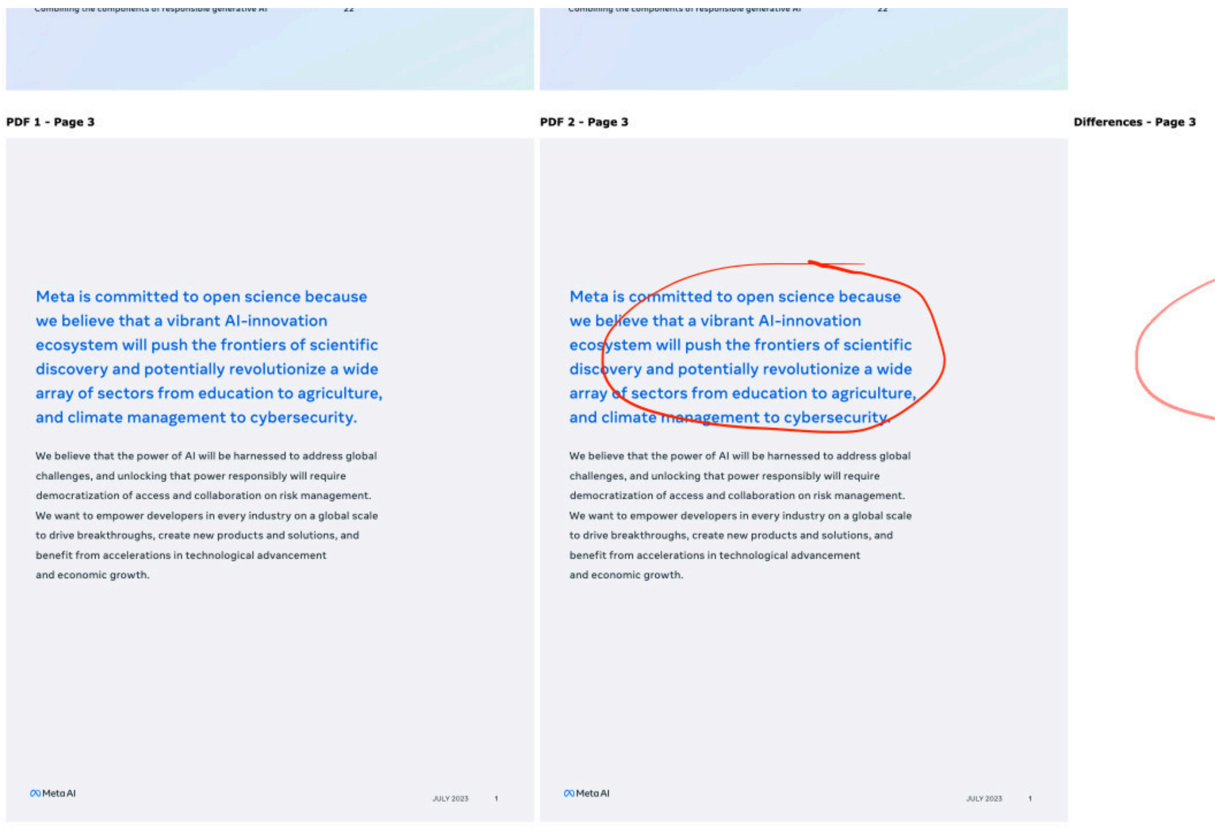
It took two prompts:

> Build a tool where I can drag and drop on two PDF files and it uses PDF.js to turn each of their pages into canvas elements and then displays those pages side by side with a third image that highlights any differences between them, if any differences exist

That give me a React app that didn't quite work, so I followed-up with this:

> rewrite that code to not use React at all

Which gave me a working tool! You can see the full Claude transcript in this Gist. Here's a screenshot of the tool in action:



Meta is committed to open science because we believe that a vibrant AI-innovation ecosystem will push the frontiers of scientific discovery and potentially revolutionize a wide array of sectors from education to agriculture, and climate management to cybersecurity.

We believe that the power of AI will be harnessed to address global challenges, and unlocking that power responsibly will require democratization of access and collaboration on risk management. We want to empower developers in every industry on a global scale to drive breakthroughs, create new products and solutions, and benefit from accelerations in technological advancement and economic growth.

Being able to knock out little custom interactive web tools like this in a couple of minutes is *so much fun*.

# 7:54 pm / pdf, projects, tools, llms, ai-assisted-programming, anthropic, claude, claude-3-5-sonnet

---

gemma-2-27b-it-llamafile (via) Justine Tunney shipped llamafile packages of Google's new openly licensed (though definitely not open source) Gemma 2 27b model this morning.

I downloaded the `gemma-2-27b-it.Q5_1.llamafile` version (20.5GB) to my Mac, ran `chmod 755 gemma-2-27b-it.Q5_1.llamafile` and then `./gemma-2-27b-it.Q5_1.llamafile` and now I'm trying it out through the `llama.cpp` default web UI in my browser. It works great.

It's a *very* capable model - currently sitting at position 12 on the LMSYS Arena making it the highest ranked open weights model - one position ahead of Llama-3-70b-Instruct and within striking distance of the GPT-4 class models.

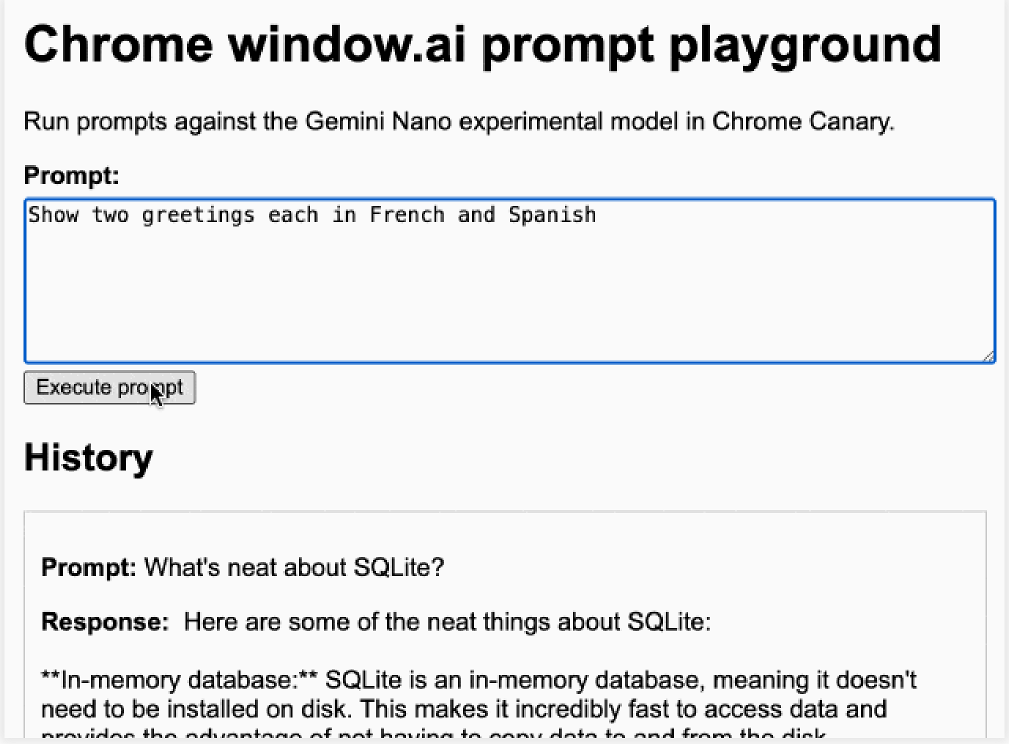
# 10:38 pm / google, ai, generative-ai, local-llms, llms, llamafile, justine-tunney, llama-cpp, gemma, chatbot-arena

---

July 3, 2024

Chrome Prompt Playground. Google Chrome Canary is currently shipping an experimental on-device LLM, in the form of Gemini Nano. You can access it via the new `window.ai` API, after first enabling the "Prompt API for Gemini Nano" experiment in `chrome://flags` (and then waiting an indeterminate amount of time for the ~1.7GB model file to download - I eventually spotted it in `~/Library/Application Support/Google/Chrome Canary/OptGuideOnDeviceModel`).

I got Claude 3.5 Sonnet to build me this playground interface for experimenting with the model. You can execute prompts, stream the responses and all previous prompts and responses are stored in `localStorage`.


Chrome window.ai prompt playground

Run prompts against the Gemini Nano experimental model in Chrome Canary.

Prompt:

```
Show two greetings each in French and Spanish
```

History

Prompt: What's neat about SQLite?

Response: Here are some of the neat things about SQLite:

In-memory database: SQLite is an in-memory database, meaning it doesn't need to be installed on disk. This makes it incredibly fast to access data and

Here's the full Sonnet transcript, and the final source code for the app.

The best documentation I've found for the new API is is explainers-by-googlers/prompt-api on GitHub.

# 5:11 pm / chrome, google, projects, ai, generative-ai, llms, ai-assisted-programming, claude, gemini

---

> If you own the tracks between San Francisco and Los Angeles, you likely have some kind of monopolistic pricing power, because there can only be so many tracks laid between place A and place B. In the case of GPU data centers, there is much less pricing power. GPU computing is increasingly turning into a commodity, metered per hour. Unlike the CPU cloud, which became an oligopoly, new entrants building dedicated AI clouds continue to flood the market. Without a monopoly or oligopoly, high fixed cost + low marginal cost businesses almost always see prices competed down to marginal cost (e.g., airlines).
>
> — **David Cahn**

# 8:49 pm / economics, ai, david-cahn, gpus

July 4, 2024

Exorcising us of the Primer (via) Andy Matuschak talks about the need for educational technologists to break free from the siren's call of "The Young Lady's Illustrated Primer" - the universal interactive textbook described by Neal Stephenson in his novel The Diamond Age.

The Primer offers an incredibly compelling vision, and Andy uses fifteen years of his own experience exploring related ideas to pick it apart and highlight its flaws.

> I want to exorcise myself of the Primer. I want to clearly delineate what makes its vision so compelling—what I want to carry in my heart as a creative fuel. But I also want to sharply clarify the lessons we *shouldn't* take from the Primer, and what it simply ignores. Then I want to reconstitute all that into something new, a vision I can use to drive my work forward.

On the Primer's authoritarianism:

> The Primer has an agenda. It is designed to instill a set of values and ideas, and while it's supportive of Nell's curiosities, those are "side quests" to its central structure. Each of the twelve "Lands Beyond" focuses on different topics, but they're not specific to Nell, and Nell didn't choose them. In fact, Nell doesn't even *know* the Primer's goals for her—she's never told. Its goals are its own privileged secret. Nell is manipulated so completely by the Primer, for so much of her life, that it's hard to determine whether she has meaningful goals or values, other than those the Primer's creators have deemed "good for her".

I'm also reminded of Stephenson's piece of advice to people who may have missed an important lesson from the novel:

> Kids need to get answers from humans who love them.

# 4:39 am / education, ai, neal-stephenson, andy-matuschak

> The expansion of the jagged frontier of AI capability is subtle and requires a lot of experience with various models to understand what they can, and can't, do. That is why I suggest that people and organizations keep an "impossibility list" - things that their experiments have shown that AI can definitely not do today but which it can **almost** do. For example, no AI can create a satisfying puzzle or mystery for you to solve, but they are getting closer. When AI models are updated, test them on your impossibility list to see if they can now do these impossible tasks.

— **Ethan Mollick**

# 10:38 pm / ethan-mollick, ai, llms

July 5, 2024

jqjq: jq implementation of jq (via) 2,854 lines of jq that implements a full, working version of jq itself. "A great way to show that jq is a very expressive, capable and neat language!"

# 3:23 pm / jq

> Product teams that are smart are getting off the treadmill. Whatever framework you currently have, start investing in getting to know it deeply. Learn the tools until they are not an impediment to your progress. That's the only option. Replacing it with a shiny new tool is a trap. [...]
>
> Companies that want to reduce the cost of their frontend tech becoming obsoleted so often should be looking to get back to fundamentals. Your teams should be working closer to the web platform with a lot less complex abstractions. We need to relearn what the web is capable of and go back to that.

— **Marco Rogers**

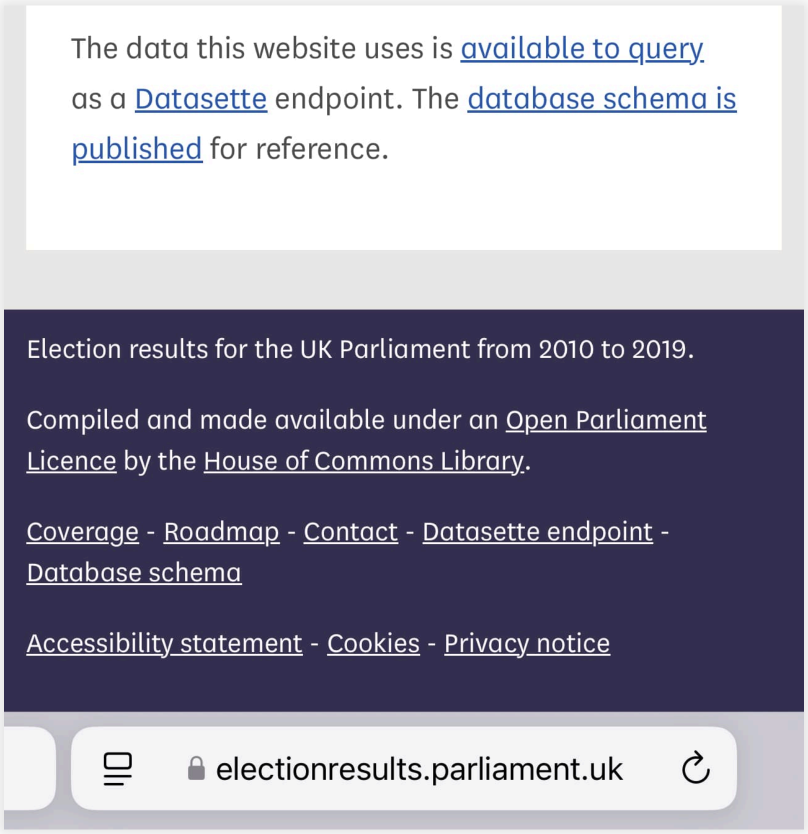
# 6:19 pm / frontend, javascript

Tracking Fireworks Impact on Fourth of July AQI (via) Danny Page ran shot-scraper once per minute (using cron) against this Purple Air map of the Bay Area and turned the captured screenshots into an animation using `ffmpeg`. The result shows the impact of 4th of July fireworks on air quality between 7pm and 7am.

# 10:52 pm / ffmpeg, shot-scraper

UK Parliament election results, now with Datasette. The House of Commons Library maintains a website of UK parliamentary election results data, currently listing 2010 through 2019 and with 2024 results coming soon.

The site itself is a Rails and PostgreSQL app, but I was delighted to learn today that they're also running a Datasette instance with the election results data, linked to from their homepage!


The data this website uses is available to query as a Datasette endpoint. The database schema is published for reference.

Election results for the UK Parliament from 2010 to 2019.

Compiled and made available under an Open Parliament Licence by the House of Commons Library.

Coverage - Roadmap - Contact - Datasette endpoint - Database schema

Accessibility statement - Cookies - Privacy notice

electionresults.parliament.uk

The raw data is also available as CSV files in their GitHub repository. Here's their Datasette configuration, which includes a copy of their SQLite database.

# 11:36 pm / elections, sqlite, datasette

---

interactive-feed (via) Sam Morris maintains this project which gathers interactive, graphic and data visualization stories from various newsrooms around the world and publishes them on Twitter, Mastodon and Bluesky.

It runs automatically using GitHub Actions, and gathers data using a number of different techniques - XML feeds, custom API integrations (for the NYT, Guardian and Washington Post) and in some cases by scraping index pages on news websites using CSS selectors and cheerio.

The data it collects is archived as JSON in the data/ directory of the repository.

# 11:39 pm / data-journalism, git-scraping, mastodon, bluesky

---

July 6, 2024

Home-Cooked Software and Barefoot Developers. I really enjoyed this talk by Maggie Appleton from this year's Local-first Conference in Berlin.

> For the last ~year I've been keeping a close eye on how language models capabilities meaningfully change the speed, ease, and accessibility of software development. The slightly bold theory I put forward in this talk is that we're on a verge of a golden age of local, home-cooked software and a new kind of developer – what I've called the barefoot developer.

It's a great talk, and the design of the slides is outstanding.

It reminded me of Robin Sloan's An app can be a home-cooked meal, which Maggie references in the talk. Also relevant: this delightful recent Hacker News thread, Ask HN: Is there any software you only made for your own use but nobody else?

My favourite version of our weird new LLM future is one where the pool of people who can use computers to automate things in their life is massively expanded.

The other videos from the conference are worth checking out too.

# 6:30 pm / ai, llms, ai-assisted-programming, local-first

July 7, 2024

Reasons to use your shell's job control. Julia Evans summarizes an informal survey of useful things you can do with shell job control features - `fg`, `bg`, `Ctrl+Z` and the like. Running `tcdump` in the background so you can see its output merged in with calls to `curl` is a neat trick.

# 4:30 pm / unix, julia-evans

July 8, 2024

> Voters in the Clapham and Brixton Hill constituency can rest easy - despite appearances, their Reform candidate Mark Matlock really does exist. [...] Matlock - based in the South Cotswolds, some 100 miles from the constituency in which he is standing - confirmed: "I am a real person." Although his campaign image is AI-generated, he said this was for lack of a real photo of him wearing a tie in Reform's trademark turquoise.

— **Private Eye**

# 3:20 pm / politics, ai, generative-ai

Geomys, a blueprint for a sustainable open source maintenance firm (via) Filippo Valsorda has been working as a full-time professional open source maintainer for nearly two years now, accepting payments on retainer from companies that depend on his cryptography Go packages.

This has worked well enough that he's now expanding: Geomys (a genus of gophers) is a new company which adds two new "associate maintainers" and an administrative director, covering more projects and providing clients with access to more expertise.

Filipino describes the model like this:

> If you're betting your business on a critical open source technology, you
>
> 1. want it to be sustainably and predictably maintained; and
> 2. need occasional access to expertise that would be blisteringly expensive to acquire and retain.
>
> Getting maintainers on retainer solves both problems for a fraction of the cost of a fully-loaded full-time engineer. From the maintainers' point of view, it's steady income to keep doing what they do best, and to join one more Slack Connect channel to answer high-leverage questions. It's a great deal for both sides.

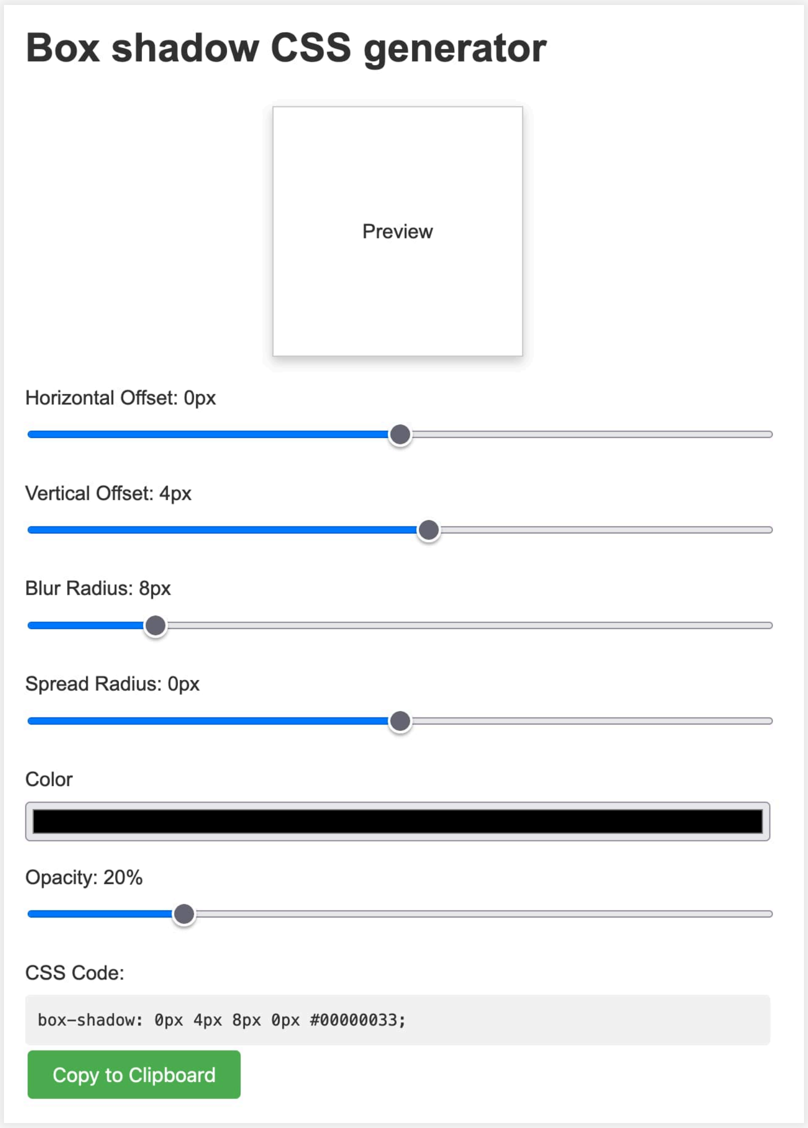
For more on this model, watch Filippo's FOSDEM talk from earlier this year.

# 3:40 pm / go, open-source, filippo-valsorda

Box shadow CSS generator (via) Another example of a tiny personal tool I built using Claude 3.5 Sonnet and artifacts. In this case my prompt was:

> CSS for a slight box shadow, build me a tool that helps me twiddle settings and preview them and copy and paste out the CSS

I changed my mind half way through typing the prompt and asked it for a custom tool, and it built me this!



Box shadow CSS generator

Horizontal Offset: 0px

Vertical Offset: 4px

Blur Radius: 8px

Spread Radius: 0px

Color

Opacity: 20%

CSS Code:

```
box-shadow: 0px 4px 8px 0px #00000033;
```

Here's the full transcript - in a follow-up prompt I asked for help deploying it and it rewrote the tool to use `<script type="text/babel">` and the babel-standalone library to add React JSX support directly in the browser - a bit of a hefty dependency (387KB compressed / 2.79MB total) but I think acceptable for this kind of one-off tool.

Being able to knock out tiny custom tools like this on a whim is a really interesting new capability. It's also a lot of fun!

# 7:30 pm / css, projects, ai, generative-ai, llms, ai-assisted-programming, anthropic, claude, claude-artifacts, claude-3-5-sonnet

---

Type click type by Brian Grubb. I just found out my favourite TV writer, Brian Grubb, is no longer with Uproxx and is now writing for his own newsletter - free on Sunday, paid-subscribers only on Friday. I hit subscribe so fast.

In addition to TV, Brian's coverage of heists - most recently Lego and an attempted heist of Graceland ("It really does look like a bunch of idiots tried to steal and auction off Graceland using Hotmail accounts and they almost got away with it") - is legendary.

I'd love to see more fun little Friday night shows too.

# 9:58 pm / blogging, tv, brian-grubb

> Someone elsewhere left a comment like "I CAN'T BELIEVE IT TOOK HER 15 YEARS TO LEARN BASIC READLINE COMMANDS". those comments are very silly and I'm going to keep writing "it took me 15 years to learn this basic thing" forever because I think it's important for people to know that it's normal to take a long time to learn "basic" things

— **Julia Evans**

# 11:15 pm / julia-evans

Jevons paradox (via) I've been thinking recently about how the demand for professional software engineers might be affected by the fact that LLMs are getting so good at producing working code, when prompted in the right way.

One possibility is that the price for writing code will fall, in a way that massively increases the demand for custom solutions - resulting in a greater demand for software engineers since the increased value they can provide makes it much easier to justify the expense of hiring them in the first place.

TIL about the related idea of the Jevons paradox, currently explained by Wikipedia like so:

> [...] when technological progress increases the efficiency with which a resource is used (reducing the amount necessary for any one use), but the falling cost of use induces increases in demand enough that resource use is increased, rather than reduced.

# 11:23 pm / wikipedia, ai, generative-ai, llms

July 9, 2024

> Inside the labs we have these capable models, and they're not that far ahead from what the public has access to for free. And that's a completely different trajectory for bringing technology into the world that what we've seen historically. It's a great opportunity because it brings people along. It gives them intuitive sense for the capabilities and risks and allows people to prepare for the advent of bringing advanced AI into the world.

— **Mira Murati**

# 3:07 am / openai, ai, llms

> Chrome's biggest innovation was the short release cycle with a silent unceremonious autoupdate.
>
> When updates were big, rare, and manual, buggy and outdated browsers were lingering for soo long, that we were giving bugs names. We documented the bugs in magazines and books, as if they were a timeless foundation of WebDev.
>
> Nowadays browser vendors can fix bugs in 6 weeks (even Safari can…). New-ish stuff is still buggy, but rarely for long enough for the bugs to make it to schools' curriculums.

— **Kornel Lesiński**

# 10:43 am / web-standards, browsers, chrome

Deactivating an API, one step at a time (via) Bruno Pedro describes a sensible approach for web API deprecation, using API keys to first block new users from using the old API, then track which existing users are depending on the old version and reaching out to them with a sunset period.

The only suggestion I'd add is to implement API brownouts - short periods of time where the deprecated API returns errors, several months before the final deprecation. This can help give users who don't read emails from you notice that they need to pay attention before their integration breaks entirely.

I've seen GitHub use this brownout technique successfully several times over the last few years - here's one example.

# 5:23 pm / apis, github

page 1 / 4 next » last »»

2024 » July

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				