August 2022

Search posts from August 2022 Search

31 posts: 5 entries, 24 links, 2 quotes

Aug. 1, 2022

storysniffer (via) Ben Welsh built a small Python library that guesses if a URL points to an article on a news website, or if it’s more likely to be a category page or /about page or similar. I really like this as an example of what you can do with a tiny machine learning model: the model is bundled as a ~3MB pickle file as part of the package, and the repository includes the Jupyter notebook that was used to train it.

# 11:40 pm / machine-learning, python, jupyter, ben-welsh

Aug. 2, 2022

How I Used DALL·E 2 to Generate The Logo for OctoSQL (via) Jacob Martin gives a blow-by-blow account of his attempts at creating a logo for his OctoSQL project using DALL-E, spending $30 of credits and making extensive use of both the “variations” feature and the tool that lets you request modifications to existing images by painting over parts you want to regenerate. Really interesting to read as an example of a “real world” DALL-E project.

# 9:12 pm / openai, dalle, generative-ai

Aug. 3, 2022

Introducing sqlite-html: query, parse, and generate HTML in SQLite (via) Another brilliant SQLite extension module from Alex Garcia, this time written in Go. sqlite-html adds a whole family of functions to SQLite for parsing and constructing HTML strings, built on the Go goquery and cascadia libraries. Once again, Alex uses an Observable notebook to describe the new features, with embedded interactive examples that are backed by a Datasette instance running in Fly.

# 5:31 pm / go, html, sqlite, datasette, alex-garcia

Aug. 4, 2022

> Your documentation is complete when someone can use your module without ever having to look at its code. This is very important. This makes it possible for you to separate your module's documented interface from its internal implementation (guts). This is good because it means that you are free to change the module's internals as long as the interface remains the same.
>
> Remember: the documentation, not the code, defines what a module does.

— **Ken Williams**

# 3:50 pm / documentation

Aug. 6, 2022

Microsoft® Open Source Software (OSS) Secure Supply Chain (SSC) Framework Simplified Requirements. This is really good: don't get distracted by the acronyms, skip past the intro and head straight to the framework practices section, which talks about things like keeping copies of the packages you depend on, running scanners, tracking package updates and most importantly keeping an inventory of the open source packages you work so you can quickly respond to things like log4j.

I feel like I say this a lot these days, but if you had told teenage-me that Microsoft would be publishing genuinely useful non-FUD guides to open source supply chain security by 2022 I don't think I would have believed you.

# 4:49 pm / microsoft, open-source, security, supply-chain

Aug. 9, 2022

sqlite-zstd: Transparent dictionary-based row-level compression for SQLite. Interesting SQLite extension from phiresky, the author of that amazing SQLite WASM hack from a while ago which could fetch subsets of a large SQLite database using the HTTP range header. This extension, written in Rust, implements row-level compression for a SQLite table by creating compression dictionaries for larger chunks of the table, providing better results than just running compression against each row value individually.

# 9:23 pm / sqlite, rust, zstd

Aug. 10, 2022

curl-impersonate (via) "A special build of curl that can impersonate the four major browsers: Chrome, Edge, Safari & Firefox. curl-impersonate is able to perform TLS and HTTP handshakes that are identical to that of a real browser."

I hadn't realized that it's become increasingly common for sites to use fingerprinting of TLS and HTTP handshakes to block crawlers. curl-impersonate attempts to impersonate browsers much more accurately, using tricks like compiling with Firefox's nss TLS library and Chrome's BoringSSL.

# 3:34 pm / crawling, curl, scraping

How SQLite Helps You Do ACID (via) Ben Johnson's series of posts explaining the internals of SQLite continues with a deep look at how the rollback journal works. I'm learning SO much from this series.

# 3:39 pm / sqlite, fly, ben-johnson

Introducing sqlite-http: A SQLite extension for making HTTP requests (via) Characteristically thoughtful SQLite extension from Alex, following his sqlite-html extension from a few days ago. sqlite-http lets you make HTTP requests from SQLite—both as a SQL function that returns a string, and as a table-valued SQL function that lets you independently access the body, headers and even the timing data for the request.

This write-up is excellent: it provides interactive demos but also shows how additional SQLite extensions such as the new-to-me "define" extension can be combined with sqlite-http to create custom functions for parsing and processing HTML.

# 10:22 pm / http, sqlite, alex-garcia

Let websites framebust out of native apps (via) Adrian Holovaty makes a compelling case that it is Not OK that we allow native mobile apps to embed our websites in their own browsers, including the ability for them to modify and intercept those pages (it turned out today that Instagram injects extra JavaScript into pages loaded within the Instagram in-app browser).

He compares this to frame-busting on the regular web, and proposes that the X-Frame-Options: DENY header which browsers support to prevent a page from being framed should be upgraded to apply to native embedded browsers as well.

I'm not convinced that reusing X-Frame-Options: DENY would be the best approach—I think it would break too many existing legitimate uses—but a similar option (or a similar header) specifically for native apps which causes pages to load in the native OS browser instead sounds like a fantastic idea to me.

# 10:29 pm / adrian-holovaty, browsers, privacy, security

Aug. 11, 2022

sethmlarson/pypi-data (via) Seth Michael Larson uses GitHub releases to publish a ~325MB (gzipped to ~95MB) SQLite database on a roughly monthly basis that contains records of 370,000+ PyPI packages plus their OpenSSF score card metrics. It's a really interesting dataset, but also a neat way of packaging and distributing data—the scripts Seth uses to generate the database file are included in the repository.

# 1:02 am / github, pypi, sqlite, seth-michael-larson

---

datasette on Open Source Insights (via) Open Source Insights is "an experimental service developed and hosted by Google to help developers better understand the structure, security, and construction of open source software packages". It calculates scores for packages using various automated heuristics. A JSON version of the resulting score card can be accessed using `https://deps.dev/_/s/pypi/p/{package_name}/v/`

# 1:06 am / open-source, security, datasette

Litestream backups for Datasette Cloud (and weeknotes)

My main focus this week has been adding robust backups to the forthcoming Datasette Cloud.

[... 1,604 words]

5:19 pm / ocr, s3, datasette, weeknotes, datasette-cloud, fly, litestream, gpt-3, dalle

---

Aug. 13, 2022

Bypassing macOS notarization (via) Useful tip from the geckodriver docs: if you've downloaded an executable file through your browser and now cannot open it because of the macOS quarantine feature, you can run "xattr -r -d com.apple.quarantine path-to-binary" to clear that flag so you can execute the file.

# 12 am / macos, security

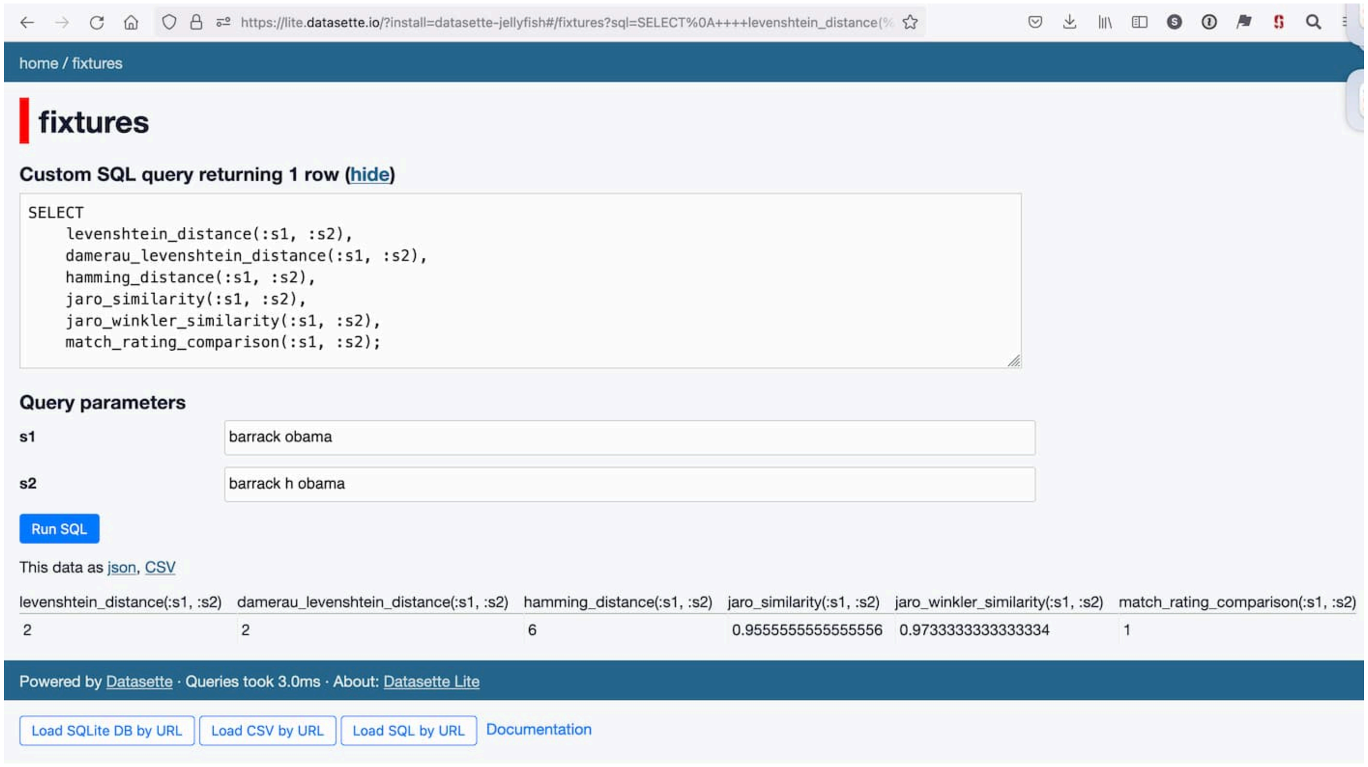
Aug. 16, 2022

Efficient Pagination Using Deferred Joins (via) Surprisingly simple trick for speeding up deep OFFSET x LIMIT y pagination queries, which get progressively slower as you paginate deeper into the data. Instead of applying them directly, apply them to a "select id from ..." query to fetch just the IDs, then either use a join or run a separate "select * from table where id in (...)" query to fetch the full records for that page.

# 5:35 pm / performance, sql

Aug. 17, 2022

Plugin support for Datasette Lite



https://lite.datasette.io/?install=datasette-jellyfish#/fixtures?sql=SELECT%0A++++levenshtein_distance(%

home / fixtures

fixtures

Custom SQL query returning 1 row (hide)

```
SELECT
    levenshtein_distance(:s1, :s2),
    damerau_levenshtein_distance(:s1, :s2),
    hamming_distance(:s1, :s2),
    jaro_similarity(:s1, :s2),
    jaro_winkler_similarity(:s1, :s2),
    match_rating_comparison(:s1, :s2);
```

Query parameters

s1	barrack obama
s2	barrack h obama

Run SQL

This data as json, CSV

levenshtein_distance(:s1, :s2)	damerau_levenshtein_distance(:s1, :s2)	hamming_distance(:s1, :s2)	jaro_similarity(:s1, :s2)	jaro_winkler_similarity(:s1, :s2)	match_rating_comparison(:s1, :s2)
2	2	6	0.9555555555555556	0.9733333333333334	1

Powered by Datasette · Queries took 3.0ms · About: Datasette Lite

Load SQLite DB by URL | Load CSV by URL | Load SQL by URL | Documentation

I've added a new feature to Datasette Lite, my distribution of Datasette that runs entirely in the browser using Python and SQLite compiled to WebAssembly. You can now install additional Datasette plugins by passing them in the URL.

[... 865 words]

6:20 pm / plugins, projects, pypi, datasette, webassembly, pyodide, datasette-lite, cors

Crunchy Data: Learn Postgres at the Playground (via) Crunchy Data have a new PostgreSQL tutorial series, with a very cool twist: they have a build of PostgreSQL compiled to WebAssembly which runs in the browser, so each tutorial is accompanied by a working psql terminal that lets you try out the tutorial contents interactively. It even has support for PostGIS, though that particular tutorial has to load 80MB of assets in order to get it to work!

# 6:30 pm / postgresql, webassembly

Building games and apps entirely through natural language using OpenAI's code-davinci model. A deeply sophisticated example of using prompts to generate entire working JavaScript programs and games using the new code-davinci OpenAI model.

# 7:06 pm / gpt-3, openai, prompt-engineering, generative-ai, llms

Aug. 19, 2022

The Datasette Newsletter: Datasette Lite, Datasette Tutorials, Datasette Cloud. It’s been quite a while since I’ve sent one of these out now—hoping to get this on to a more regular schedule.

# 1:20 am / datasette

Aug. 20, 2022

Shoelace (via) Saw this for the first time today: it’s a relatively new library of framework-agnostic Web Components, built on lit-html and covering a huge array of common functionality: buttons and sliders and dialogs and drawer interfaces and dropdown menus and so on. The design is very clean, the documentation is superb—and it looks like you can cherry pick just the components you are using for a pretty lean addition to your page weight. So refreshing to see libraries like this that really take advantage of modern web standards.

# 8:57 pm / css, javascript, web-standards, web-components, lit-html

---

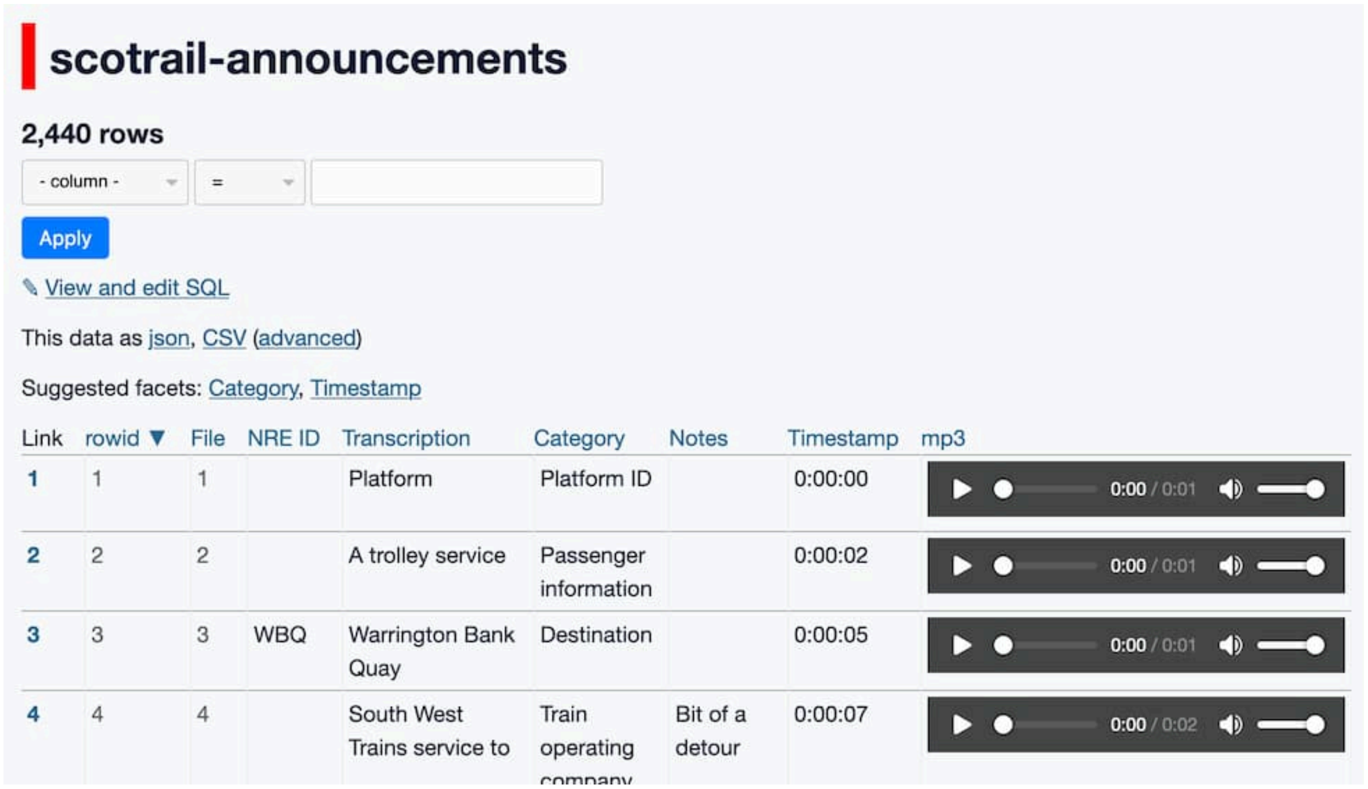
Show HN: A new way to use GPT-3 to generate code (and everything else). Riley Goodside is my favourite Twitter follow for GPT-3 tips. Here he describes a powerful prompt pattern he's designed which lets you generate extremely complex code output by asking GPT-3 to fill in `$$areas like this$$` with different patterns, then stitch them together into full HTML or other source code files. It's really clever.

# 9:33 pm / gpt-3, prompt-engineering, generative-ai, riley-goodside, llms

---

Aug. 21, 2022

Analyzing ScotRail audio announcements with Datasette—from prototype to production


scotrail-announcements

2,440 rows

This data as json, CSV (advanced)

Suggested facets: Category, Timestamp

Link	rowid ▼	File	NRE ID	Transcription	Category	Notes	Timestamp	mp3
1	1	1		Platform	Platform ID		0:00:00	0:00 / 0:01
2	2	2		A trolley service	Passenger information		0:00:02	0:00 / 0:01
3	3	3	WBQ	Warrington Bank Quay	Destination		0:00:05	0:00 / 0:01
4	4	4		South West Trains service to	Train operating company	Bit of a detour	0:00:07	0:00 / 0:02

Scottish train operator ScotRail released a two-hour long MP3 file containing all of the components of its automated station announcements. Messing around with them is proving to be a huge amount of fun.

[... 4,428 words]

2:04 am / projects, datasette, datasette-lite

Turning SQLite into a distributed database (via) Heyang Zhou introduces mvSQLite, his brand new open source “SQLite-compatible distributed database” built in Rust on top of Apple’s FoundationDB. This is a very promising looking new entry into the distributed/replicated SQLite space: FoundationDB was designed to provide low-level primitives that tools like this could build on top of.

# 5:40 pm / databases, sqlite, rust

Aug. 22, 2022

Digitizing 55,000 pages of civic meetings (via) Philip James has been building public, searchable archives of city council meetings for various cities—Oakland and Alamedia so far—using my s3-ocr script to run Textract OCR against the PDFs of the minutes, and deploying them to Fly using Datasette. This is a really cool project, and very much the kind of thing I’ve been hoping to support with the tools I’ve been building.

# 4:26 pm / archiving, ocr, political-hacking, datasette, fly

Stable Diffusion Public Release (via) New AI just dropped. Stable Diffusion is similar to DALL-E, but completely open source and with a CC0 license applied to everything it generates. I have a Twitter thread (the via) link of comparisons I’ve made between its output and my previous DALL-E experiments. The announcement buries the lede somewhat: to try it out, visit beta.dreamstudio.ai—which you can use for free at the moment, but it’s unclear to me how billing is supposed to work.

# 7:12 pm / machine-learning, dalle, stable-diffusion, generative-ai, text-to-image

Aug. 24, 2022

How SQLite Scales Read Concurrency (via) Ben Johnson’s series on SQLite internals continues—this time with a detailed explanation of how the SQLite WAL (Write-Ahead Log) is implemented.

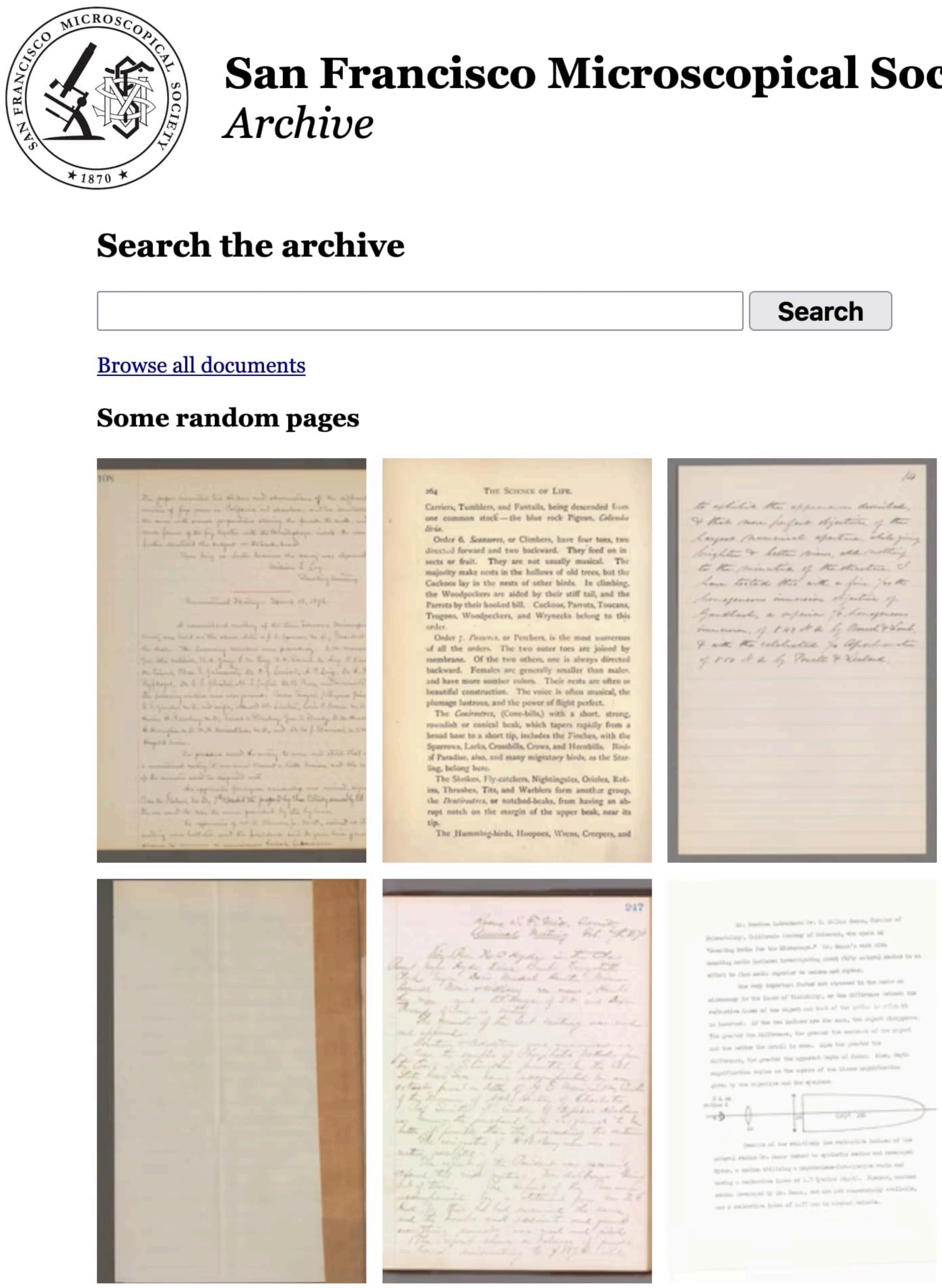
# 4:16 pm / databases, sqlite, ben-johnson

> To make the analogy explicit, in Software 1.0, human-engineered source code (e.g. some .cpp files) is compiled into a binary that does useful work. In Software 2.0 most often the source code comprises 1) the dataset that defines the desirable behavior and 2) the neural net architecture that gives the rough skeleton of the code, but with many details (the weights) to be filled in. The process of training the neural network compiles the dataset into the binary — the final neural network. In most practical applications today, the neural net architectures and the training systems are increasingly standardized into a commodity, so most of the active “software development” takes the form of curating, growing, massaging and cleaning labeled datasets.
>
> — **Andrej Karpathy**

# 9:28 pm / data, machine-learning, ai, andrej-karpathy

Aug. 25, 2022

Building a searchable archive for the San Francisco Microscopical Society



San Francisco Microscopical Society
Archive

Search the archive

Search

Browse all documents

Some random pages

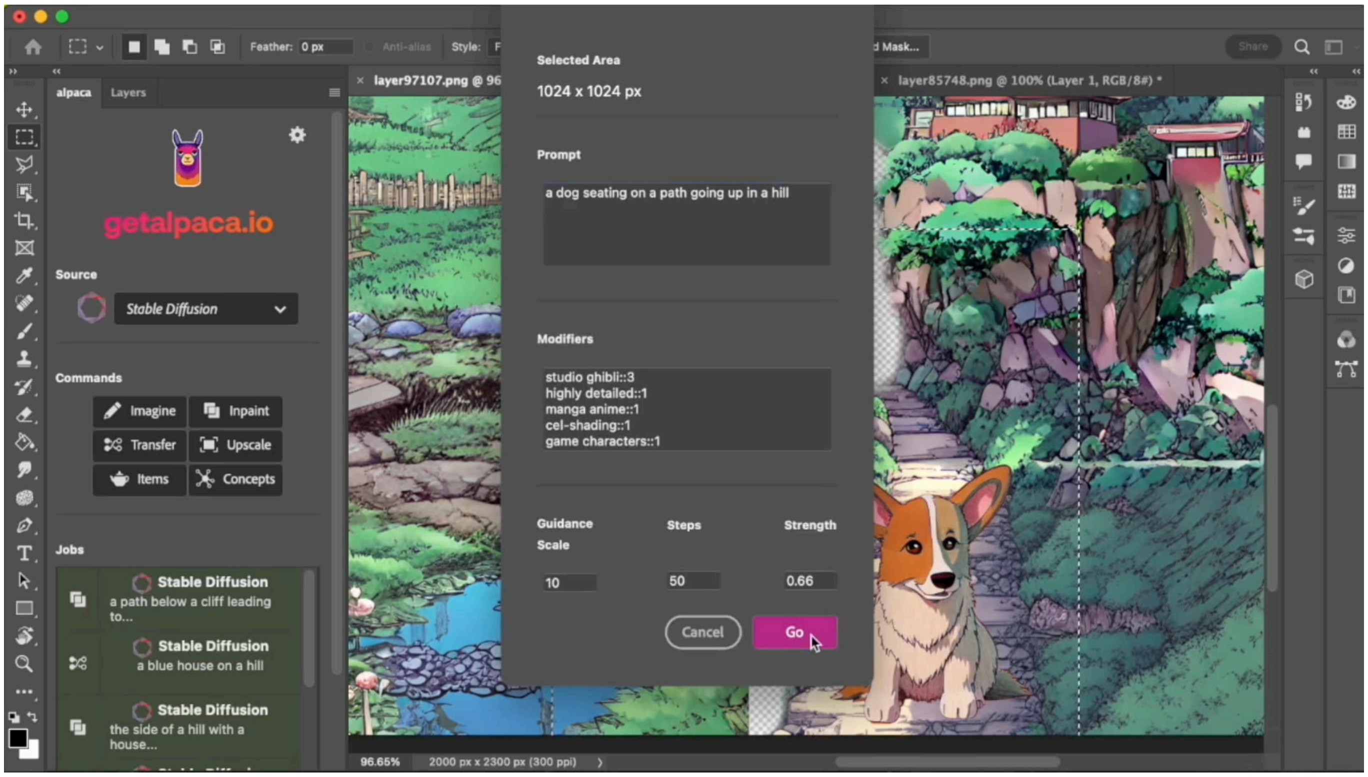
The San Francisco Microscopical Society was founded in 1870 by a group of scientists dedicated to advancing the field of microscopy.

[... 1,845 words]

5:24 pm / ocr, pdf, projects, datasette, weeknotes

Aug. 29, 2022

Stable Diffusion is a really big deal



Selected Area
1024 x 1024 px

Prompt
a dog seating on a path going up in a hill

Modifiers
studio ghibli::3
highly detailed::1
manga anime::1
cel-shading::1
game characters::1

Guidance Scale: 10
Steps: 50
Strength: 0.66

Source: Stable Diffusion

Jobs
Stable Diffusion
a path below a cliff leading to...
Stable Diffusion
a blue house on a hill
Stable Diffusion
the side of a hill with a house...

If you haven't been paying attention to what's going on with Stable Diffusion, you really should be.

[... 1,443 words]

1:09 am / ethics, machine-learning, ai, dalle, stable-diffusion, prompt-engineering, generative-ai, laion, text-to-image, ai-ethics

Aug. 31, 2022

Exploring 12 Million of the 2.3 Billion Images Used to Train Stable Diffusion's Image Generator. Andy Baio and I collaborated on an investigation into the training set used for Stable Diffusion. I built a Datasette instance with 12m image records sourced from the LAION-Aesthetics v2 6+ aesthetic score data used as part of the training process, and built a tool so people could run searches and explore the data. Andy did some extensive analysis of things like the domains scraped for the images and names of celebrities and artists represented in the data. His write-up here explains our project in detail and some of the patterns we've uncovered so far.

# 2:10 am / machine-learning, ai, stable-diffusion, generative-ai, laion, training-data

page 1 / 2 next »

2022 » August

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				