

JustHTML is a fascinating example of vibe engineering in action

14th December 2025

I recently came across [JustHTML](#), a new Python library for parsing HTML released by Emil Stenström. It's a very interesting piece of software, both as a useful library and as a case study in sophisticated AI-assisted programming.

First impressions of JustHTML

I didn't initially know that JustHTML had been written with AI assistance at all. The README caught my eye due to some attractive characteristics:

- It's pure Python. I like libraries that are pure Python (no C extensions or similar) because it makes them easy to use in less conventional Python environments, including Pyodide.
- "Passes all 9,200+ tests in the official [html5lib-tests](#) suite (used by browser vendors)"—this instantly caught my attention! HTML5 is a big, complicated but meticulously written specification.
- 100% test coverage. That's not something you see every day.
- CSS selector queries as a feature. I built a Python library for this [many years ago](#) and I'm always interested in seeing new implementations of that pattern.
- html5lib has been [inconsistently maintained](#) over the last few years, leaving me interested in potential alternatives.
- It's only 3,000 lines of implementation code (and another ~11,000 of tests.)

I was out and about without a laptop so I decided to put JustHTML through its paces on my phone. I [prompted Claude Code for web](#) on my phone and had it build [this Pyodide-powered HTML tool](#) for trying it out:

Playground Mode

CSS Selector Query

Pretty Print HTML

Tree Structure

Stream Events

Extract Text

To Markdown

CSS Selector:

p

Run Query

Output

3 matches

```
<p class=intro>
  A
  <strong>pure Python</strong>
  HTML5 parser.
</p>
```

This was enough for me to convince myself that the core functionality worked as advertised. It's a neat piece of code!

Turns out it was almost all built by LLMs

At this point I went looking for some more background information on the library and found Emil's blog entry about it: [How I wrote JustHTML using coding agents](#):

Writing a full HTML5 parser is not a short one-shot problem. I have been working on this project for a couple of months on off-hours.

Tooling: I used plain VS Code with Github Copilot in Agent mode. I enabled automatic approval of all commands, and then added a blacklist of commands that I always wanted to approve manually. I wrote an [agent instruction](#) that told it to keep working, and don't stop to ask questions. Worked well!

Emil used several different models—an advantage of working in VS Code Agent mode rather than a provider-locked coding agent like Claude Code or Codex CLI. Claude Sonnet 3.7, Gemini 3 Pro and Claude Opus all get a mention.

Vibe engineering, not vibe coding

What's most interesting about Emil's 17 step account covering those several months of work is how much software engineering was involved, independent of typing out the actual code.

I wrote about [vibe engineering](#) a while ago as an alternative to vibe coding.

Vibe coding is when you have an LLM knock out code without any semblance of code review—great for prototypes and toy projects, definitely not an approach to use for serious libraries or production code.

I proposed “vibe engineering” as the grown up version of vibe coding, where expert programmers use coding agents in a professional and responsible way to produce high quality, reliable results.

You should absolutely read [Emil's account](#) in full. A few highlights:

1. He hooked in the 9,200 test [html5lib-tests](#) conformance suite almost from the start. There's no better way to construct a new HTML5 parser than using the test suite that the browsers themselves use.
2. He picked the core API design himself—a TagHandler base class with handle_start() etc. methods—and told the model to implement that.
3. He added a comparative benchmark to track performance compared to existing libraries like html5lib, then experimented with a Rust optimization based on those initial numbers.
4. He threw the original code away and started from scratch as a rough port of Servo's excellent [html5ever](#) Rust library.
5. He built a custom profiler and new benchmark and let Gemini 3 Pro loose on it, finally achieving micro-optimizations to beat the existing Pure Python libraries.
6. He used coverage to identify and remove unnecessary code.

7. He had his agent build a [custom fuzzer](#) to generate vast numbers of invalid HTML documents and harden the parser against them.

This represents a lot of sophisticated development practices, tapping into Emil's deep experience as a software engineer. As described, this feels to me more like a lead architect role than a hands-on coder.

It perfectly fits what I was thinking about when I described **vibe engineering**.

Setting the coding agent up with the html5lib-tests suite is also a great example of [designing an agentic loop](#).

“The agent did the typing”

Emil concluded his article like this:

JustHTML is about 3,000 lines of Python with 8,500+ tests passing. I couldn't have written it this quickly without the agent.

But “quickly” doesn't mean “without thinking.” I spent a lot of time reviewing code, making design decisions, and steering the agent in the right direction. The agent did the typing; I did the thinking.

That's probably the right division of labor.

I couldn't agree more. Coding agents replace the part of my job that involves typing the code into a computer. I find what's left to be a much more valuable use of my time.

Posted [14th December 2025](#) at 3:59 pm · Follow me on [Mastodon](#), [Bluesky](#), [Twitter](#) or [subscribe to my newsletter](#)

More recent articles

- [I ported JustHTML from Python to JavaScript with Codex CLI and GPT-5.2 in 4.5 hours](#) - 15th December 2025
- [OpenAI are quietly adopting skills, now available in ChatGPT and Codex CLI](#) - 12th December 2025

html 94

python 1214

ai 1738

generative-ai 1535

llms 1500

ai-assisted-programming 289

vibe-coding 59

coding-agents 111

Next: [I ported JustHTML from Python to JavaScript with Codex CLI and GPT-5.2 in 4.5 hours](#)

Previous: [OpenAI are quietly adopting skills, now available in ChatGPT and Codex CLI](#)

Monthly briefing

Sponsor me for **\$10/month** and get a curated email digest of the month's most important LLM developments.

Pay me to send you less!

Colophon © 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016
2017 2018 2019 2020 2021 2022 2023 2024 2025