jordanhubbard/nanolang (via) Plenty of people have mused about what a new programming language specifically designed to be used by LLMs might look like. Jordan Hubbard (co-founder of FreeBSD, with serious stints at Apple and NVIDIA) just released exactly that.

> A minimal, LLM-friendly programming language with mandatory testing and unambiguous syntax.
>
> NanoLang transpiles to C for native performance while providing a clean, modern syntax optimized for both human readability and AI code generation.

The syntax strikes me as an interesting mix between C, Lisp and Rust.

I decided to see if an LLM could produce working code in it directly, given the necessary context. I started with this MEMORY.md file, which begins:

> **Purpose:** This file is designed specifically for Large Language Model consumption. It contains the essential knowledge needed to generate, debug, and understand NanoLang code. Pair this with `spec.json` for complete language coverage.

I ran that using LLM and llm-anthropic like this:

```
llm -m claude-opus-4.5 \
  -s
https://raw.githubusercontent.com/jordanhubbard/nanolang/re
\
  'Build me a mandelbrot fractal CLI tool in this
language'
  > /tmp/fractal.nano
```

The resulting code... did not compile.

I may have been too optimistic expecting a one-shot working program for a new language like this. So I ran a clone of the actual project, copied in my program and had Claude Code take a look at the failing compiler output.

... and it worked! Claude happily grepped its way through the various `examples/` and built me a working program.

Here's the Claude Code transcript - you can see it reading relevant examples here - and here's the finished code plus its output.

programming-languages 41 ai 1790 generative-ai 1583 llms 1549 ai-assisted-programming 309 llm 247 coding-agents 127 claude-code 76

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!

Sponsor & subscribe

I've suspected for a while that LLMs and coding agents might significantly reduce the friction involved in launching a new language. This result reinforces my opinion.

Posted 19th January 2026 at 11:58 pm

Recent articles

- First impressions of Claude Cowork, Anthropic's general agent - 12th January 2026
- My answers to the questions I posed about porting open source code with LLMs - 11th January 2026
- Fly's new Sprites.dev addresses both developer sandboxes and API sandboxes at the same time - 9th January 2026