

10 Months with Supermaven in Neovim

Karl Matthias

Member of Technical Staff - Mozi



supermaven



neovim

Who Am I

Nearly 30 years in tech, 10 startups

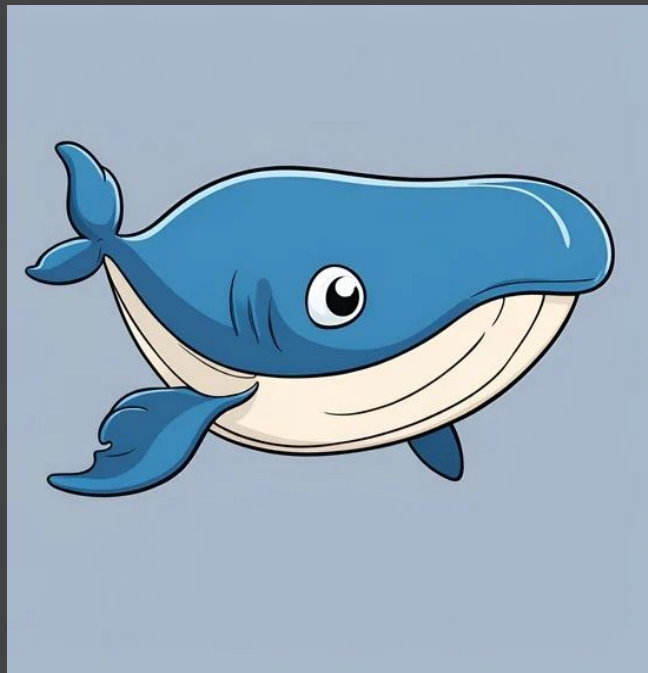
ex-New Relic, ex-Nitro, ex-InVision,
ex-Community.com (etc)

Co-Wrote *Docker: Up and Running* from
O'Reilly Media (3 editions)

Working remote since 2018

In Ireland since April 2016

Coding every day



This Talk

What I've learned from daily use

Practical use of the tool

Why I made the choices I did

Reflects previous 10 months — things are moving fast

Mostly about how to use it, not how it works

Largely about using AI *auto-complete* to full advantage



Why I Use AI

I code a lot of the day, ~5 hours a lot of days

More time to focus on big ideas and important stuff (like code structure, test strategy)

I am less fatigued

Less tedium

Replaces some other tools



Auto-Complete vs Chat

Used for different things

Work together

Use two different models

Have two different contexts (but should they?)



Auto-Completion

Where you spend 95% of your time

Must be Fast!

Must be (fairly) correct!

No need to supply reasoning

Should primarily be good at edits

Knows your codebase! (with enough context)



Chat

Great for starting things, large chunks of code

For reference

Maybe for refactors—but not great yet
IME



My Dev Tools Stack

Neovim — Editor

Supermaven — Auto-completion model
and editor integration

ChatGPT — Chat



Pairing with AI

This is a pairing session!

Tell it what you want to do

Get the right context!

Think about what it should know

Bigger context is better here

Supermaven vs Copilot — 1 million
tokens vs 8192



Pairing with AI

Big enough codebase gives more context, better results (to a point)

Intentionally open files in the editor from outside the repo that give context

Use a well-known language with lots of code examples. e.g. Go, Ruby vs Crystal

Load any outside schemas if you have them (DB, protobuf, etc)



Demo 🙌

Tips Summary

Make sure it has the right context!

Comments to generate code

Test description to generate tests

Partially edit a line as much as you can remember. Re-edit to let it fix it.

Make the same change back-to-back for best context.



Where to Find Me

Blog: relistan.com

Mastodon: [@relistan@k.matthias.org](https://k.matthias.org/@relistan)

LinkedIn:

<https://www.linkedin.com/in/karlmatthias/>

Book: *Docker: Up and Running (3rd ed)*

Mozi: <https://mozi.app>

