

Anecdotally, programmers dislike "reduce"

by [Evan Hahn](#), posted Sep 13, 2026, tagged [#software-development](#) [#culture](#)

In short: from my experience, people like `map` and `filter`, but not `reduce`.

I use functions like `map` and `filter` all the time. When I put that code up for review, my peers rarely complain. I get plenty of feedback about other decisions, but not about my use of `map` and `filter`.

I cannot say the same for `reduce`. Often, when I've submitted a patch with `reduce` inside, I get a comment like, "this part is hard to read." And I see `reduce` way less than `map`, `filter`, `some`, and so on.

Anecdotally, I have come to believe that programmers don't like `reduce` as much.

I don't know why, but I have a few theories:

- `reduce` is harder to read.
- `reduce` is less familiar.
- `reduce` *can* have worse performance compared to other options.
- `reduce` *is* less elegant in languages I use, like JavaScript, Python, and Swift. In my blissful stint as a Clojure developer, I did *not* get this feedback.
- I'm wrong, and I'm seeing a trend that's not real.

I usually just change `reduce` to something else and move on. Even though I prefer it, I don't usually care much. But it's a little social phenomenon I've observed, and I thought I'd document it.

I've also noticed this less recently, possibly because code review is less thorough nowadays.

Do you notice this? Do you like `reduce`? Please tell me.



[About me](#) [Contact](#) [Projects](#) [Guides](#) [Blog](#)

[RSS](#) [Newsletter](#) [Mastodon](#)

Unless noted otherwise, content is licensed under the [Creative Commons Attribution-NonCommercial License](#) and code under the [Unlicense](#). Logo by [Lulu Tang](#). Profile photo by [Ali Boschert-Downs](#).