

Reinventing the wheel

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — [77e1b28a](#)

1 Question

Why do we reinvent the wheel?

2 Answer

It is very common for developers to want to develop things themselves instead of reusing existing code. They prefer to know how it works so that if they need to make changes to the code, they know where to do it since they wrote it themselves. If they are lucky, they've been given a narrow set of initial requirements that could be fulfilled by a more complex/complete library, but they think that they can do it themselves or do it better. Thus they build themselves a [square wheel](#).

One of the positive sides of developing something that already exists is that you can get a good understanding of what is required without having to consider all the other parts which may be required for other use cases. You're also free to explore the aspect of the problem that you find interesting or challenging.

One of the negative sides is that if you work on your own square wheel for a while, you never end up learning a library that might prove more useful. You also do not acquire the skill of learning new libraries and their strengths/weaknesses, as well as how you could help to improve them. Sometimes however by reading an existing library you will come to the conclusion that fixing said library is likely to take you more time than writing a new one.

In my experience, the main reason that has led me to write something from scratch even though I had access to code that partially did what I wanted was because:

- the library wasn't able to sell me on the reasons I should use it instead of doing it myself, while I considered implementing the solution myself to be somewhat easy.
- the code (organization, style) was so messy that fixing it would require more time than rewriting it after understanding the core concepts.
- the library was not maintained anymore.
- using the library would make my workflow slower.

3 References

- https://en.wikipedia.org/wiki/Reinventing_the_wheel