Reusable writing

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — 861fb9d0

I'm a programmer. I like to write code and whenever I can, I like to reuse bits of code I've written in the past instead of writing it from scratch.

Lately I've been spending a lot of my time thinking. However, unlike programming, I feel that my writing is repetitive and that I sometimes don't make any progress with my ideas. Oh how I wished writing was more like programming.

There's many ways one can write. You have the top-down approach, where you first lay out a plan of the topics you want to write about. You also have the bottom-up approach, where you write things as they come to mind, sometimes rearranging the content. Generally, when I write, it is to dump what is on my mind about a certain topic, so it is very much like the bottom-up approach. However, the thing I dislike about it is that you can end up with subtle similar sections of content, but not know about it. You actually need to go through all of your writing to discover these bits of duplicate writing. That's a lot similar to writing code and sometimes finding out you have functions/methods that do almost the same thing...

In programming, when you realize that two bits of code are similar, you refactor your code to get one out in order to reduce duplication. When you write, it's not as simple. First off, in code you will generally have encapsulation, which provides you with the means of limiting the scope of the duplicate logic. With text, your context (encapsulation) often comes from the text prior to it.

One option to deal with this problem might be to extract ideas and then rewrite them in order to make them independent of their previous context. This will help making it possible to read any part of an article without having read what was said previously. However, this will also have the adverse effect of making things much more difficult to follow as there will appear to be no connection between the different sections.

Maybe a better approach to the problem is to think of how one makes code more reusable. One of the biggest issues with code is often that multiple levels of abstractions are used within the same method, which makes reasoning about the method more difficult.

1 Questions

- Should one limit the size of an idea, in the same fashion one tries to limit the number of lines or statements in a function?

2 References

- http://wiki.c2.com/?AlexandrianForm
- https://www.martinfowler.com/articles/writingPatterns.html
- https://en.wikipedia.org/wiki/Pattern_language
- https://www.jacquimalpass.com/lick-it-you-can-reuse-your-knowledge/