

Typst finds a sweet spot for taking scientific notes

John Doe · August 19, 2025

There are already plenty of note taking tools, and many of them have more or less support for taking notes of scientific thoughts. The bad news is that most of them, if not all, do not meet the criteria of taking scientific notes.

1. Taking scientific notes need modular reusable snippets for mathematics

Scientific notes usually contains mathematical contents that are typeset in some language, most commonly LaTeX. As the note base grows, shared snippets would appear in clusters of notes. These snippets, if not made reusable as macros or functions in the typesetting language, will become unmaintainable just like how a program without usage of variables and functions are unmaintainable. In the meantime, making all the snippets accessible globally risks leakage of abstraction and namespace pollution. Hence, these typesetting snippets should act just like definitions in programming languages: reusable by referring to their name, and scoped under a module system.

2. Taking scientific notes needs full-power mathematical typesetting

Most of the note-taking software, especially those based by web techniques, has limited support for LaTeX simulation through MathJax or KaTeX. However, these simulations never satisfy one who tries to has any fine-grained control over mathematical typesetting. Anyone who wish to typeset mathematics in their notes seriously would need the full-power of it. This is not necessarily achieved using LaTeX, but at least one needs to use something that declares itself to be a full-power solution.

3. Markdown is not suitable for scientific content producing

Given that Taking scientific notes need modular reusable snippets for mathematics and Taking scientific notes needs full-power mathematical typesetting, it is obvious that Markdown, without extensions, does not satisfy any of the two principles, and any of the software-specific extensions that tries to solve these problems does not really solve them, just causing migration friction. This is because mathematics is alien to Markdown — it is just not part of it, thus there cannot be effective integration.

4. Pure LaTeX is not suitable for scientific content producing

Many might argue against this, but I think that pure LaTeX, that is, just LaTeX itself, is not suitable for scientific content producing. What I agree is that LaTeX is suitable for serious scientific content *publishing*. In that scenario, fine-grained control over typesetting is important, and following the popular standard and utilizing the ecosystem is also important. LaTeX is probably the best one for that purpose. However, I think there are more stuff besides publishing in scientific content producing — one needs to effectively take notes of scientific contents, create new scientific contents not only for publishing (for example, private note, temporary thought, manuscript, etc.), and manage these contents. And the following disadvantages prevent LaTeX to be suitable for these purpose: its compilation is too slow, thus one cannot preview their input and reflect upon it immediately; its language is far from modern, thus it is hard for one to be able to grasp it so that they can build their own utilities on it; also, it is too heavy to be integrated into other tools.

Typst, as a typesetting software with a domain-specific script language, or as a script language with typeset content as one of its basic value type and output target, has the ability to create modular, reusable snippets with enough control over typesetting. Its markup language and script language, by both producing content as values in the language, are congruent at a higher level. And its instant-preview is really impressive.