Markdown 3 at TUG 2023: Reflections from the Q&A session

Vít Starý Novotný

Abstract

In my TUG 2023 talk, I unveiled the third iteration of the Markdown package for TEX. Following the talk, a number of intriguing questions were raised by attendees, which remained open-ended at that time. This article seeks to address the questions in depth.

1 Introduction

At TUG 2023, I presented the third major version of the Markdown package for TEX. A recording as well as presentation slides are available online [1].

Following the talk, three intriguing questions were raised by attendees:

1. What are the advantages of markdown compared to LaTeX?

2. Does the Markdown package support markdown dialects such as GitHub Flavored Markdown?

3. Did I prepare my presentation slides using the Markdown package?

However, due to time constraints, the questions remained open-ended at the time of the conference. In this article, I address the questions more thoroughly and provide the comprehensive answers they deserve.

2 The dubious advantages of markdown compared to LaTeX

In the first question of the Q&A session, an audience member expressed doubts about the need for a yet another markup language. Namely, they questioned the readability benefits of using punctuation instead of LaTeX commands to denote document structure.

Punctuation was historically used to indicate structure in manuscripts. Markdown's adoption of punctuation adheres to this time-tested convention:

```
# Exotic Fruits Galore!
Discover these *unique fruits*:
- Rambutan: Hairy, red, sweet.
- Jackfruit: Large, sweet pods.
- Salak: Scaly, tangy-sweet.
```

By contrast, LaTeX employs commands articulated in natural language, predominantly English:

```
\section{Exotic Fruits Galore!}
Discover these \emph{unique fruits}:
\begin{itemize}
\item Rambutan: Hairy, red, sweet.
\item Jackfruit: Large, sweet pods.
\item Salak: Scaly, tangy-sweet.
\end{itemize}
```

Although LaTeX commands are more conspicuous than punctuation, they infuse English elements into non-English documents, which hinders readability, especially to non-English speakers.

During my talk, I presented new syntax extensions supported by the Markdown package such as task lists, superscripts, and subscripts. The audience member pointed out that each new syntax extension added more special characters, which would make the markdown language less predictable to authors.

In the Markdown package, each syntax extension must be deliberately activated. Therefore, when a new syntax extension is introduced to the package, it impacts authors only if they choose to activate it. Furthermore, if authors have portions of their document written without the new extension and are concerned that activating the syntax extension may alter their document, they can conveniently activate the extension for selected content only:

```
% Activate extension only in chapter 2.
\markdownInput              {chapter-1.md}
\markdownInput[newExtension]{chapter-2.md}
\markdownInput              {chapter-3.md}
```

3 Support for markdown flavors other than CommonMark

In the second question of the Q&A session, an audience member remarked that activating syntax extensions for tables makes the language recognized by the Markdown package close to GitHub Flavored Markdown (GFM). They asked whether we planned to further support GFM in the Markdown package.

The Markdown package supports the CommonMark dialect of markdown. Other dialects of markdown such as MultiMarkdown and PHP Markdown Extra have subtle differences in parsing various elements and are not supported by the Markdown package. If you have documents written in other dialects of markdown, you can use a conversion tool such as Pandoc to convert your documents to CommonMark.

For example, here is how you would convert a MultiMarkdown document with tables to CommonMark using Pandoc from the command line:

```
pandoc -i original.md -f markdown_mmd \
 -t commonmark+pipe_tables -o converted.md
```

You would then typeset the converted document using the Markdown package as follows:

```
\markdownInput[pipe_tables]{converted.md}
```

Although GFM is also a dialect, it is a strict superset of CommonMark. Therefore, documents written for the Markdown package with no syntax extensions activated are always valid GFM documents.

Vít Starý Novotný

Furthermore, most extra GFM elements compared to CommonMark such as tables, task lists, and strikethrough are supported by the Markdown package as syntax extensions. Therefore, GFM documents with no other extra elements are valid in the Markdown package with the `pipe_tables`, `task_lists`, and `strike_through` syntax extensions activated.

In GitHub Docs, GFM also supports YAML metadata, which are valid in the Markdown package with the `jekyll_data` syntax extension activated.

If support for the remaining extra elements of GFM [2] is added to the Markdown package in the future, we may add a convenience option `gfm` that would enable all syntax extensions of GFM.

4 Writing presentation slides with the Markdown package for TₑX

In the third question of the Q&A session, an audience member asked whether I used the Markdown package to prepare my presentation slides [1].

Markdown can be used to write many different kinds of structured documents, such as books, articles, and technical documentation. Although presentation slides are also structured, they lean towards the visual and somewhat informal style evident in posters and leaflets.

To prepare my presentation slides, I used the WYSIWYG editor of Google Spreadsheets. This allowed me to not only structure the textual content but also to craft the layout of each individual slide.

If your presentation slides are more structured and less visual, you can write them in markdown. Examples of using the Markdown package for presentation slides are given in my previous *TUGboat* article [3, Section 1.2] and elsewhere [4, 5, 6].

5 Conclusion

In my TUG 2023 talk, I unveiled the third iteration of the Markdown package for TₑX. My talk generated a series of thoughtful questions from attendees during the ensuing Q&A session. In this article, I revisited the questions and answered them in depth.

At the time of the conference, Markdown 3 was still in development. A stable version of Markdown 3 was released in August [7, 8, 9] and will be included in TₑX Live 2024.

References

[1] V. Starý Novotný. Markdown 3: What's new, what's next, July 2023. `youtu.be/U8XjTOhJkg0` and `tug.org/tug2023/program.html`

[2] Github flavored markdown spec, Apr. 2019. Version 0.29-gfm. `github.github.com/gfm`

[3] V. Novotný. Markdown 2.10.0: LᴬTₑX themes & snippets, two flavors of comments, and LuaMetaTeX. *TUGboat* 42(2):186–193, Sept. 2021. `https://tug.org/TUGboat/tb42-2/tb131novotny-markdown.pdf`

[4] V. Novotný. One size does not fit all, Sept. 2021. `overleaf.com/read/mfpwbywkmgnv`

[5] V. Novotný. When tesseract brings friends, Dec. 2021. `overleaf.com/read/gsmdcmrmxrmr`

[6] V. Novotný. Interpretable representations for fast and accurate retrieval of mathematical information, 2022. `github.com/witiko/doctoral-thesis/blob/main/presentation.tex`

[7] V. Starý Novotný, P. Olšák, A. Genčur. Witiko/markdown: Release 3.0.0-alpha, 2023. `github.com/witiko/markdown/releases/tag/3.0.0-alpha`

[8] V. Starý Novotný. Witiko/markdown: Release 3.0.0-alpha.2, 2023. `github.com/witiko/markdown/releases/tag/3.0.0-alpha.2`

[9] V. Starý Novotný, L. Prentice, M. Peischl, D. Willis, T. Vrabcová. Witiko/markdown: Release 3.0.0, 2023. `github.com/witiko/markdown/releases/tag/3.0.0`

⋄ Vít Starý Novotný
  Studená 453/15
  Brno, 638 00
  Czech Republic
  `witiko (at) mail dot muni dot cz`
  `github.com/witiko`

Figure 1: Markdown wolf writes a letter to audience members. Illustration by `fiverr.com/quickcartoon`.