About 1400 Words of Skepticism about Markdown, and an Imagined Alternative

Chris Forster

2015-06-29

`http://cforster.com/2015/06/markdown-skepticism`

Don't get me wrong, Markdown's great. Indeed, nearly all the writing I do now is in Markdown (or at least starts that way). There has been a good amount of writing about the virtues of Markdown for academic writing in particular, so I'll just link to them here:

- W. Caleb McDaniel's Why (and How) I Wrote My Academic Book in Plain Text
- Nikola Sander's Writing Academic Papers in Markdown Using Sublime Text and Pandoc
- Dennis Tennen and Grant Wythoff's Sustainable Authorship in Plaintext Using Pandoc and Markdown (This latter I especially recommend).

But Markdown, as it stands, has some drawbacks, which become acute when you are trying to extend it to cover the needs of academic writing (or, say, as a transcription format for texts).

The Problem

What I will describe as "problems" all stem from the fact that Markdown remains essentially a simplified syntax for HTML. A tool like Pandoc, which has a special (and especially powerful) flavor of Markdown all its own, helps reduce the borders between document formats. With Pandoc it becomes easy to convert `HTML` to `LaTeX`, or Rich Text Format to Word's `.docx`. It could easily feel like Markdown is a universal document format—write it in Markdown, and publish as whatever.

That is a lovely dream—an easy-to-write plaintext format that can easily be output to any desired format. In reality, though, Markdown (even Pandoc's Markdown) remains yoked to HTML, and so it suffers from some of its problems.

The problem I encounter most frequently in HTML (and in Markdown) concerns nesting a block quote within a paragraph. In short, can you have a block quote *within* a paragraph? If you're writing HTML (or MarkDown), the answer is no—HTML treats "block quotes" as `block` *elements*; this means that one cannot be contained within a paragraph (this restriction does not exist in LaTeX or TEI). Yet, what could be more common in writing on works of literature? Representing poetry presents its own problems for HTML and Markdown.By contrast to the challenge presented by the mere fact of poetry, note the many syntaxes/tools available for fenced code blocks, syntax highlighting, and so on; Markdown, for now, remains of greatest interest to software developers and so reflects their habits and needs.(*Note*: If you're looking for practical advice, you can easily represent poetry in Pandoc's markdown using "line blocks"; this is not a perfect solution, but it will do for many needs).

Perversely, markdown also represents something of a step backward with regard to *semantics*. If you've spent some time with HTML, you may have noticed how HTML5 cements a model of HTML as a semantic markup language (with, implicitly, matters of presentation controlled by CSS). That means that the `<i>` tag,

which long ago meant *italics*, has since acquired semantic meaning. According to the w3c, it should be used to "represent[] a span of text offset from its surrounding content without conveying any extra emphasis or importance, and for which the conventional typographic presentation is italic text; for example, a taxonomic designation, a technical term, an idiomatic phrase from another language, a thought, or a ship name." Those instances where one wishes to express emphasis, use the `<em>` tag. If you need to mark a title, don't simply italicize it, use `<cite>` .But hold up, that `cite` element obscures the distinctions we normally make between italicizing certain titles and putting others in quotation marks. In practice, of course, I doubt these distinctions are widely respected across the web; but all those at least *potentially* useful distinctions are lost in markdown, whose syntax marks them all with `*` or `_`. Markdown is, in fact, rather *unsemantic*. (To a lesser degree, one might detect this tendency as well in the way headings—rather than `divs`—are Markdown's primary way of structuring a document, but I'll stop now.) So, two points: Markdown inherits HTML's document which includes an inability to nest block-level elements within paragraphs; in simplifying HTML, it produces a less semantically clear and rich format. (Technically, of course, one could simply include any HTML element for which Markdown offers no shortened syntax—like `<cite>` for example.)

A Solution

On the CommonMark forum, some folks have proposed additional syntax to fix the latter problem, and capture some of the semantic distinctions mentioned above (indeed, following the discussions over there has helped sensitize to me some of the challenges and limitations of markdown as a sort of universal format donor). So, some of these issues could be resolved through extensions or modifications of Markdown.

Yet, given these deficits in Markdown, I wonder if it isn't worth ask-

ing a more basic question—whether the plaintext format for "academic" writing should be so tightly yoked to HTML? If Markdown is, fundamentally, a simplified, plaintext syntax for HTML, could we imagine a similar, easy-to-write, plaintext format that wouldn't be tied to HTML? Could we imagine, say, a format that would represent a simplification of syntax, not of HTML, but of a format better suited to the needs of representing more complex documents? Could we imagine a plaintext format that would be to TEI, say, what markdown is to HTML?

Such a format would not need to *look* particularly different from Markdown. Its syntax could overlap significantly; as in Pandoc's Markdown format, file metadata (things like title, author, and so on) could appear (perhaps as YAML) at the front of the file (and be converted into elements within `teiHeader`). You could still use `*`, `**`, and `[]()` as your chief tools; footnotes and references could be marked the same way (you could preserve Pandoc's wonderful citation system, with such things represented as `<refs>` in TEI).

The most substantive difference would not be in syntax, but in the document model. Any Markdown file can contain HTML—all HTML is valid markdown; this ensures that Markdown is never less powerful than HTML. But are the burdens of HTML worth the costs if one wishes to do scholarly/academic, or similar types of writing, in plaintext? Projects exist to repurpose Pandoc markdown for scholarly writing: Tim T. Y. Lin's ScholarlyMarkdown, or Martin Fenner's similar project, or the workflow linked-to above, by Dennis Tennen and Grant Wythoff at the Programming Historian. What I'm imagining, though, is entirely less practical than any of these projects at the moment because it would necessitate a change in the document model into which markdown is converted. Pandoc works its magic by reading documents from a source format (through a "reader") into an intermediary format (a format of its own that you can view by outputting `-t native`), which it can then output (through a "writer"). Could TEI (or some representation of it), essentially, fulfill that role as intermediary

format? (A Pandoc car with a TEI engine swapped in?)

I like writing in plaintext, but I don't love being bound by the peculiarities that Markdown has inherited from HTML. So, it is worth considering what it is that people like about Markdown. I suspect that most of the things people like about Markdown (free, easy to write, nonproprietary, easily usable with version control, and so on), have little to do with its HTML-based document model but stem from its being a plaintext format (and the existing infrastructure of scripts/apps/workflows around markdown). TEI provides an alternative document model—indeed, a *richer* document model. Imagine a version of Pandoc that uses TEI (or a simplified TEI subset) behind the scenes as its native format. Folks often complain about the complexity and verbosity of TEI (and XML more generally), and not without reason. I would certainly never want to *write* TEI; but a simplified TEI syntax that could then take advantage of all the virtues of TEI, that would be something.

[Closing Note: At one point I wondered how easy it would be to convert markdown to TEI with Pandoc… I've managed to finagle a set of scripts to do that; it's janky, but for anyone interested, it's here.]