More notes on notes (sidenotes, endnotes, footnotes, etc)

Scott Laird

2024-12-17 2024-12-18 About 2000 words 9 minutes

So, yesterday's article on sidenotes in Hugo led me down yet another rabbithole, and I'm still not quite sure where it leads.

I want to add notes to my blog and have them usually displayed in the right margin alongside the text. I'd like them to be usable on mobile devices, which don't really *have* a right margin. I'd like them to be easy to write, without a lot of extra work or friction. And I'd like them to be supportable long-term, not needing a pile of bad hacks or a complicated processing pipeline to get them to render.

Why sidenotes

I'm generally a fan of Tufte, and he has a nice summary of the advantages of side (or margin) notes vs footnotes on his site. His focus is mostly on letter-sized paper, but his arguments hold even better with desktop websites – there's just too much width to be useful for readable text, and putting notes in the margin works better than putting them at the end of the page because (a) the space wasn't being used anyway and (b) they're closer to the user's eye and easier to find.

The driving limitations here are (a) font size and (b) characters per line for maximizing readability. According to Wikipedia, the optimal line length for readability is 45-75 characters per line, with the ideal around 66 characters. With proportionally-spaced English text, US letter-sized paper, and 1" margins, that's somewhere around 15 point text, which is pretty large. Spot-checking a couple docs that I have in Google Docs, it looks like its default of 11 point Arial gives me 85-100 characters per line. Effectively, the margins are too small for optimal readability unless the font size is bumped way up.

Because of this, sidenotes are *usually* better than endnotes or footnotes for the web or full-sized paper when readability is a goal.

Mobile

However, this *doesn't* really hold for smaller formats, like paperback books or phones. They're running in to the opposite problem, where text either needs to be tiny or it's has too few words for optimal readability.

So, if you want to add notes to a blog post, what should you do for optimal readability?

For desktop formatting, this is easy: side (or margin) notes. Put the note in the margin. The right side makes more sense to me, but I've seen left work well enough; in print they'll usually be on the outside edge which will alternate between the left and right margin anyway.

For mobile (or smaller screens in general), though? That's more difficult. I've seen 4 different approaches:

1. Turn them into endnotes at the bottom of the page. Harder to read the notes but easier to read the actual doc.
2. Put them inline between paragraphs, usually formatted differently (in a box, different background) so that they stand out.
3. Hide them by default and have the user click on an indicator (the footnote number, etc) to make them appear.
4. Don't show the notes on mobile devices.

I should probably mention this long writeup of the various note options before I go much further; it has examples of all of these.

I haven't seen a single example of #2 or #3 that actually looks okay for general-purpose writing. I'm sure it's possible, especially for things like technical reference works, but every example just looks *cluttered*, and clicking on small items on mobile isn't very easy for a lot of devices.

So far here I've resorted to option #4: just don't show notes. It's not ideal, but if the notes were *critical* then they wouldn't be notes, they'd be part of the main text.

But the ideal, at least to me, is probably #1: turn the side notes into either footnotes or endnotes on mobile.

Not everyone agrees with this, of course. Martin Fowler doesn't want his notes interfering with the reader's flow through the document. In that case, foot/end notes are probably ideal, although *actually finding and reading the endnotes* is a much bigger distraction than reading sidenotes in a lot of cases. This is largely personal preference, although I suspect that careful formatting will make sidenotes more useful most of the time.

Printing

As an aside, formatting for printing opens up a whole extra can of worms. See this paged.js page that rather exhaustively covers the topic. For footnotes, you really want them to appear *as footnotes* on each page, not endnotes at the end of the document, which is what natively printing HTML will give you. You can probably fix that with JavaScript, but *in general* formatting HTML for optimal printing isn't easy. I'm going to punt on this for now, but may come back after I get mobile working.

Markers or not?

So, how do you indicate which notes go with which text? And do you even need to? It all depends on what you're doing with your notes.

The Tufte CSS people (and presumably Tufte himself) distinguish between “sidenotes” and “margin notes”; sidenotes are numbered like footnotes, while margin notes are un-numbered and just kind of hang out in the margin near the text that they’re associated with.

If you’re just adding a couple brief asides, then un-marked margin notes are probably fine. On the other hand, if you’re explicitly clarifying points in the text or using lots of notes (where alignment gets tricky), then numbered notes are going to be clearer.

Molly White’s `annotate` framework goes even further and wraps numbered notes around blocks of text, ala Google Docs’ comment system. I like a few parts of this, but for my uses it’s overkill. I’d love to be able to click on sidenotes and have the text associated with it highlighted, which `annotate` can do, but I *don’t* want text highlighted by default (it’s less readable) and there’s no good system that I can see for marking text and typing it to comments inside of Markdown. I debated adapting `annotate` and hiding the annoying bits via CSS, but the usability inside of Markdown makes it too much trouble. For more heavily annotated things (like her writing) with commentary from multiple authors, it’s probably a great solution.

My first attempt at sidenotes didn’t use any sort of marker, but I quickly discovered that some of the comments were confusing because it wasn’t clear which text they belonged with. So today’s version uses numbered footnote-like markers.

Usability when writing

Ignoring the output format, what should notes look like *when writing*, especially when using Markdown? Under the hood, it’s all just HTML, but no one really wants to write a bunch of `<span class="sidenote"><small>My text</small></span>` tags every time you create a note. Besides being annoying to type and hard to read, it also doesn't really cover the *intent* of what I'm trying to do. I want to say "I have a note" and have notes get formatted correctly, not muck around with formatting directly in my text. In an ideal world, how should notes be written?

As I wrote yesterday, I've been using Hugo's "shortcode" feature to wrap notes up in `{{< sidenote >}}` tags. It works, but it's a bit hard to read when you drop multi-sentence notes into a paragraph. Hugo's shortcodes are better for standardizing how inline images or embedded Tweets look than as a text formatting helper.

While basic Markdown doesn't include footnote support, a lot of Markdown implementations include an extension for footnote formatting. These include Github-flavored Markdown and Goldmark, the Markdown implementation that Hugo uses by default.

I'd much rather say:

```
Text text[^1] more text...

[^1]: Footnote
```

or

```
Text text ^[Footnote] more text...
```

than

```
Text text {{< sidenote >}}Footnote{{< /sidenote >}} more text...
```

So in an ideal world, I could just format my text using extended Markdown footnotes and get everything rendered as sidenotes. Unfortunately, Goldmark's footnote formatting isn't really conducive to this. It drops all of the footnote text in a `<div>` at the bottom of the page, which makes them difficult to use as sidenotes – they're in the wrong place in the HTML stream to be able to easily move them back up with the text.

Fortunately, Hugo is flexible, and it includes at least 5 different Markdown-to-HTML renderers by default. One of these, Pandoc has a more flexible footnote formatter *and* has a reasonably common extension for turning footnotes into sidenotes. All you need to do is install Pandoc and pandoc-sidenote and tell Hugo to use them. Since both Pandoc and pandoc-sidenote are included in Ubuntu, all you need is `apt install pandoc pandoc-sidenote`. Then configure configure Hugo to add `--filter pandoc-sidenote` to the Pandoc command line. Which is *slightly* tricky because Hugo doesn't actually have a control for this; I needed to create a `pandoc` shell script that added the filter flag and then called the "real" `pandoc`.

I'll write up the details for this in another post later, along with the CSS that I'm using.

Wrapping things up

Using `pandoc-sidenotes` and a little bit of CSS seems to produce good-enough results on desktop, but I'd really prefer to use endnotes on mobile, and I don't see a good way to accomplish this as-is. The ideal fix would probably be to modify `pandoc-sidenotes` to add sidenotes but leave the original footnotes in place as endnotes with CSS tags to indicate which ones were moved to sidenotes. Unfortunately, `pandoc-sidenotes` is written in Haskell, so that might be a much bigger project than I'd like. Presumably on desktop we can hide the endnote version of sidenotes while on mobile we can hide the sidenotes and leave the endnotes intact. It'll make pages slightly larger, but I don't have *that* much side content, so I don't think it'll matter in practice.

The only good alternative that I see is to use Javascript to either move footnotes to sidenotes or sidenotes to footnotes; neither one is very appealing to me.

For now, I'm just going to leave the notes hidden on mobile.

Update 2024-12-18: Moving from Goldmark to Pandoc seems to have had one small side-effect that I'd missed initially – it broke code syntax highlighting. I've cleaned it up a bit for now, but it's in a kind of intermediate state – not great, not terrible. I like the way Goldmark was rendering code much better, but I *don't* like the way that notes look with it, and it doesn't currently support Goldmark's short footnote syntax (`text^[footnote]`). So, do I add short footnote syntax to Goldmark, or do I live with somewhat worse code formatting and figure out how to get mobile sidenotes in Pandoc? Goldmark is in Go, which is easy for me, while Pandoc is Haskell, which I've never written and is kind of notoriously difficult to get started with. But, neither Hugo nor Goldmark support runtime extensions, while Pandoc does. So with Pandoc I can just write an external filter and be done with things, while with Goldmark I'd need to create a (very minor) fork of Hugo or get my changes merged upstream.

Updated on 2024-12-18

#sidenotes #blog #typography