Why We Should Have Markdown Rendered Websites

You're viewing this document in your HTML-rendering browser but its source is actually a markdown file.

And if you've ever worked in the web development industry, you must be aware of the myriad frameworks, services and processes put in place that let people publish text. Major newsoutlets like the New York Times, bloggers on Substack and Wikipedia authors - all of them have this one thing in common: They publish text to the internet via intermediary software that automates formatting and distribution.

Similarly, if we zoom into the work of front end developers, we're finding that the depth of required knowledge to ship a reasonably-usable website has become non-trivial in an endless sea of browser-specific-rendering and new device ratios being added daily.

In fact, the guild of webdevs have become so upset and cynical about it that code-golfing a minimal "looks good everywhere" configuration is now a popular and controversial challenge [1].

But this problem is being attacked from all sides - the browser vendor's too: with many of them now integrating a "reader mode" to de-clutter busy websites to the minimum. It's what everybody wants: Continuously readable text, inline images/videos, special inline formatting and linking.

So I think it's clear that this sentiment is being shared on the user and vendor side: Cluttered HTML pages are a real problem for everyone. And while the existing solutions have done a decent job in addressing the problems, the space's general trajectory is one that acknowledges fragmentation and doubles down on publishers' liberty towards freedom in presentation.

But I'm not here to argue for that in this letter today. In fact, I'm here to argue for the opposite, for a future browser modus that takes on the role of an opinionated presenter: One that grants the user a reasonable space for expression (e.g. textual formulation, integration of multimedia etc) - but that defines the medium's limits contrarily to how we've done it in the past:

Stopping to frame the browser as a "user-hosted and abstracted computing platform" towards a presentation-independent data format for text.

I think both can co-exist and I don't want to argue away the legit use cases of HTML, CSS and JS. But with HTML5 and since I've professionally joined the movement in 2015, the tools have had such great escape velocity and complexity has increased so dramatically: We really need a new publishing tool for everyone that doesn't require intricate knowledge of how the entire web computing stack functions.

And so if we want to keep the commercialization of this domain in check: the myriad of blogging software, static HTML generators and other tools, then I think having browsers to integrate near-perfect rendering of `.md` files for readability can be a reasonable and simple-to-learn alternative to the HTML, CSS, JS stack.

Consider the following two files a simple website anyone could set up in minutes:

```
// file: http://home.md

[home](http://home.md) [about](http://about.md)

this my homepage

// file: http://about.md

[home](http://home.md) [about](http://about.md)

this my about page
```

And then for comparisson, consider this page's source. I'm a professional working for years on end in this field. This website is an expression of my mild-perfectionism and bias towards simplicity. I've had to pay close attention to make this website's source code simple but simultaneously broadly-readable on all device ratios. It's not trivial knowledge and certainly not reproducible by a person merely wanting to publish multimedia-containing texts online.

But today, at least the browser that I use, surpringly don't pretty-render markdown files and so they can technically not be used to build a simple web presence. Curiously though, we have support for other types of files: Music can be reproduced, videos too. We have WebGL and whatever other types of media being supported natively in all browsers.

Pretty-rendering markdown files would suddenly enable everyone to express themselves broadly and with full browser and device ratio support. Something that today only a commercial service or a professional web developer can offer.

It is clear to me that there's a market for this. From the vendor's perspective with them building these "reader modes" into the software, and from a user/publisher's perspective: as all we've been doing is writting markdown-flavored texts to eventually publish them as HTML.

What I've now described, I see it mostly as a coordination problem between browser vendors. Honestly, I'm sceptical any change can happen soon - but I guess I was optimistic enough about the potential for change given that I wrote and published this document.

Let's have markdown rendering in all major browsers soon. Happy to help where I can. Any help, feedback and introductions are appreciated.

Best, Tim Daubenschütz tim@daubenschuetz.de

References

- 1: https://gist.github.com/JoeyBurzynski/617fb6201335779f8424ad9528b72c41