Frank Seifferth

On why this website is a pdf file

12th April 2023

It has been common knowledge for quite some time that everything web is pretty bloated. To make matters even worse, it is usually saturated with lots of ads, user tracking and gatekeeper corporations. The current web is also highly dynamic and extremely volatile. The amounts of javascript that are loaded to display even the simplest of pages are ever increasing, requiring ever greater amounts of memory and processing power on the client side and degrading the user experience; especially for users relying on slightly aged hardware. Want to display that page without connecting to google fonts and youtube account services? Good luck! Want to download a simple blog post so you can read it while being offline? Or store an interesting piece of writing for later reference? You may as well get a degree in software engineering in order to learn how to debug that html/css/js nightmare yourself.[1] In short, the web has long stopped being a democratizing force for the free exchange of ideas. Instead it has moved on to become

[1]: I am well aware that debugging websites has nothing to do with actual computer science; but try to explain that to one of your non-programmer friends and you will quickly realize that they would really need to take that course in order to understand what you mean.

an ad-driven marketing machinery where most people just sign up with one of the gatekeepers and contend themselves with publishing their thoughts in walled gardens.

Is this the end of the story? I certainly hope not. And so do others. A couple of years back, an anonymous James made a convincing [2] argument that publishing pdf rather than html pages could actually solve most of these issues. If you are interested in the whole argument, just take a look at the original reasoning in issue 0 of the lab6 zine (available at https://lab6.com/0). The points that really drove the argument home for me, however, are the following ones:

1. PDFs are single, self-contained files using (for the most part) a stable and well defined file format. As such they can be downloaded, stored, shared. Some even claim that they could possibly be printed out on paper, although that idea does seem to have a somewhat 20th-century-ish touch to it.

2. PDFs can be read using a wide range of existing software. They can even be rendered directly by any of the major web browsers,[3] which makes them the perfect drop-in replacement for html websites. No need to install anything beside what already comes pre-installed on practically any personal computer you can get your hands on. The entry barrier for accessing content published in pdf is extremely low.

3. PDFs can also be authored using a wide range of existing software. If you enjoy laying out your content in html and css, you can use weasyprint to produce a version that would be fit for publishing on the web. If you prefer to use a Turing-complete markup language, latex is your friend

---

[2]: Opinions differ, of course, but at the very least he managed to convince me.

[3]: Which is to say, mostly, that they are supported by Chrome. They are also supported by Firefox, for that matter, and by that other version of Chrome that is branded with a slightly updated logo of Microsoft's deprecated Internet Explorer.

(although that friendship tends to be one of those tense, complicated ones). Even if you are an entirely non-technical person, you will probably be familiar with at least one piece of word processing software that allows you to create pdfs. Starting your own blog is as easy as writing a post, exporting it as pdf and copying it to a web server.

• • •

To sum up: PDF-based web technology is actually a thing. It is already ubiquitous in today's computing landscape and it provides a slow-paced, more user-friendly alternative to the ad-driven mainstream. All that is missing is its adoption by authors. This is one more website joining those ranks.

Colophon

This blog post was first published on my personal website at https://tilde.club/~seifferth/. Feel free to share it under the terms of the Creative Commons Attribution-ShareAlike License (CC BY-SA 4.0).

In order to prove that pdfs can be fun, and akin to the general spirit of the lab6 publication, this particular file also happens to be its very own LuaTeX source code. It is also a self-extracting resource bundle, for that matter, as well as a valid php script that can parse the file's LuaTeX code in order to produce a dramatic printout of this text on stdout. To compile this file to itself, change the filename extension to '.tex' and then run

```
$ luatex filename.tex
```

To watch the dramatic printout of this blog post on your terminal, simply invoke

```
$ php filename.pdf
```