lab6

Official sponsor of 6

In This Issue

Find the disk image as a PDF attachment

£1.99 JANUARY 2021

1LAB6ABnKev8dXGpHPCctyJfnUunJs13ah

ISSUE 0

Deurbanising the Web

> "I've migrated to HTML 4."
>
> - Mark Pilgrim, January 2003.

> "The software industry is currently going through the "disposable plastic" crisis the physical world went through in the mid-20th century (and is still paying down the debt for). You can run software from 1980 or 2005 on a modern desktop without too much hassle, but anything between there and 2-3 years ago? Black hole of fad frameworks and brittle dependencies. Computer Archaeology is going to become a full-time job."
>
> - Linux Weekly News, August 2020

Computing in the 2020s is (still) a user-hostile shifting sand land[1]. We are drowning in *churn* and *noise*. I am fighting back by switching this website from HTML to PDF.

Yes, blasted, accursed PDF, of all things!

"Why? Why?! For the love of all that is holy, why?!" you say!

Certainly not because it's a *great* format. PDF has many shortcomings. But in its cold, immutable fixity, it stands in opposition to the mercenary, dynamic web of rubbish. PDF makes a stand against the churn.

> "We choose to switch to PDF in this decade, not because it easy, but because it is hard"
>
> – John F. Warnock, September 12th 1962

I'm publishing this document in PDF because:

- PDFs are **self-contained** and **offlineable** – you can archive them and be confident they will remain stable and readable in the future, with no external dependencies to manage. You can't kill a PDF by shutting down an API.
- PDFs are **files**. We must not lose sight of the fact that *files are a basic freedom*. In a world where information is increasingly locked up behind APIs, files represent control for the user. A file is a sequence of bytes, of a known length. It is completely under the control of the user. A vendor cannot change it sneakily. You can checksum it and

1 Not an original idea – such complaints are commonplace. Here's one, and here's another, and another, and another, and another.

manage its integrity. You can sign it and manage its authenticity. You can back it up and distribute it easily. You can sneakernet it and samizdat it. You can parse it and convert it to another format. You can work with it offline. We have 60 years of tooling available to manage files.

- PDFs are **decentralised**. You may have obtained this PDF from a website, or maybe not! Self-contained static files are liberating! They stand alone and are not dependent on being hosted by any particular web server or under any particular domain. You can publish to one host or to a thousand hosts or to none, and it maintains its identity through content-addressing, not through the blessing of a distributor.
- PDFs are **discoverable**. Search engines index them as easily as any other format.
- PDFs are **independent** of browsers – but can still be read easily by most browsers.
- PDFs and a PDF tool ecosystem **exist today**. No need for another ghost town GitHub repo with a promising README and v0.1 in progress. PoC||GTFO!
- PDF is an **open standard**, which is freely available[2], and *stable*. It has a version number and many interoperable implementations including free and open source readers and editors.
- PDFs are **part of the web**, as much as HTML – they support hyperlinks and you can link to specific pages with URL fragments.
- PDFs are **page-oriented**. This is another fundamental freedom – to know unambiguously which part of the document you are looking at. Compare to infinite-scroll HTML pages which are disorienting by design. This may sound trivial, but seriously: with infinite scrolling, you are fundamentally not in control of the reading experience. Let's ignore the fact that most infinite scrolling implementations are janky. The problem is that you have no clear notion of when the document will end, whether you will be permitted to reach the end, whether the end will still be available by the time you get there, or what hurdles (adverts) lie in your path. It is harder to reference a section, to share it, to bookmark it, to download it, and to find it again. You are reliant on the site to inform you where you are and where you need to go next. Unsurprisingly, this disorientation is a feature, from the perspective of

2 Although it is a scandal that ISO want 198 Swiss Francs for a copy of the PDF 2.0 spec, ISO 32000-2:2017

the content controller. A page is a physical unit of information just as the paragraph is a logical unit of information. HTML focuses on logical structure and leaves the physical structure to the browser, which ostensibly formats it for optimum viewing by the user, but in practice formats it for optimum "engagement" with the advertiser. Without pages, users are adrift.

PDF's historical disadvantages are now either resolved or mitigated:

- PDFs used to be large, and although they are still larger than equivalent HTML, they are still an order of magnitude smaller than the torrent of JavaScript sewage pumped down to your browser by most sites[3].
- PDFs used to be inaccessible, but now you can tag them.
- PDFs used to be unreadable on small screens, but now you can reflow them.
- PDFs used to be read-only, but now there are tools to edit them[4], or dump their contents to other formats.
- PDFs used to be best consumed as printed pages, but now there is an abundance of PDF tooling for viewing and manipulating them on screens of all sizes.

How did it come to this?

Some say the organic web is dead, but it *is* still out there. It's *out there* in the same sense that the open countryside is still out there: local, boutique, and characterful, but gradually thinning in favour of a great migration away to the cities. Web denizens now spend their online lives in the immense arcologies of Facebook, Google, Twitter, Reddit, the walled gardens of app stores, and the walled fortresses of corporate networks. Basic infrastructure is gloriously taken care of by the City – no need to be concerned with what lies beneath the streets. It's all just there waiting to serve eager customers. Plush furnishings and smooth highways. All you need to do is go when the light turns green and stop when the light turns red. You'll never be lost because everywhere is a destination. All this, for only $\$\overline{9}.99$.

The countryside has been paved.

3 The http archive State of the Web report for July 2020 claims the median desktop page size is 2000KB, across 70 requests.
4 But it's still best if you don't.

In the city, you are free to walk the streets and gawp through the windows, urged not to notice you are on somebody else's property, subject to their terms of service (you did read them, didn't you?), consuming that service using devices they approve, at a time they decide, in a format conducive to monetisation, and in an order set by an algorithm tuned by addiction psychologists to maximise your spend. There is a one-way system in force: the downward scroll.

Content marketing has infected the web. Advertising has funded many wonderful things, but with the disastrous side-effect of poisoning the web's semantics. Every topic has been SEO'd, and everyday searches lead to pages of fake plastic human-interest, the length and shape of which has been designed solely around the multiple megabytes of adverts, distractions, upsells, misdirections, invitations to sign up to newsletters, cookie warnings, GDPR warnings… and *maybe* some content darting out of view as another ad loads. Is it unfair to blame advertising for the government-mandated cookie warnings? Not at all, because websites don't *need* to use tracking cookies, and they only need a GDPR warning if they process personal data. *Quit processing my personal data!*

Articles are undated in an attempt to stay evergreen. If information architecture had a Geneva Convention, this would be a war crime.

The churn is relentless – content is changed without notice or record of history, or republished under different domains. The search engines[5] are choked of their oxygen supply, with every search keyword assailed by a zombie horde of have-a-go algorithmically-generated content farms, trying to out-pagerank each other and exhausting the namespace around those keywords.

The tracking and monitoring is all-pervasive. Sites that failed to get a click out of you try to wring some residual value from your visit through harvesting your personal data, or maybe even by hammering your CPU and battery with some cryptomining JavaScript.

Mobile is the worst, hemmed in by hustlers on all sides, with only a narrow blinkered proprietary viewport to look through, and the bonnet sealed shut.

5 Both of 'em!

Follow the money

The thermonuclear cash generator at the heart of AdTech is driven by *engagement* and *attention*. The more time users spend on your site, the more adverts you can show them. And what should the ad vendors do when every web page is completely saturated by advertising? Easy! Expand the scope of the web to include every possible mode of content creation and consumption and in fact all of computing in its entirety, thereby attracting more eyeballs!

Starting with Web 2.0 techniques such as XMLHttpRequest, which was the key enabling technology for reloadless single-page-applications, we now have a web of applications, not just a web of pages. The web is no longer just HTML, but also Web GL, Web Sockets, Web Assembly, Web RTC, Web NFC, Web Storage, a Vibration API… the catalogue of specs is monstrous.

Drew DeVault demonstrates it hilariously:

> *"The total word count of the W3C specification catalogue is 114 million words at the time of writing. If you added the combined word counts of the C11, C++17, UEFI, USB 3.2, and POSIX specifications, all 8,754 published RFCs, and the combined word counts of everything on Wikipedia's list of longest novels, you would be 12 million words short of the W3C specifications.*
>
> *I conclude that **it is impossible to build a new web browser**. The complexity of the web is obscene. The creation of a new web browser would be comparable in effort to the Apollo program or the Manhattan project.*
>
> *It is impossible to:*
>
> - *Implement the web correctly*
> - *Implement the web securely*
> - *Implement the web **at all**"*

The old web standards have been co-opted; the W3C has capitulated and accepted that HTML is now ruled by the WHATWG. Who are they? The browser vendors!

Now, browsers today are incredibly sophisticated and powerful. The browser engineers deserve immense respect for the technology they have created. We live in a web application paradise thanks primarily to the efforts (and $billions) of Mozilla and Google. Web applications are neat, but the browser has become the all-consuming gatekeeper of interaction and engagement. Driven by this demand to expand its domain of control, the browser is now so complex that it requires web-advertising-scale investment to maintain. The browser vendors have achieved devops apotheosis in their rapid release cadence, but have done so out of

necessity, driven by endless scope creep. The recent interest in VR and AR is fuelled by a dollar-signs-for-eyeballs desire for a *terra nullius* to pave with billboards.

> *"Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones which can."*
>
> - Zawinski's Law

When is a browser finished? ChromeOS's answer is: not until it has completely taken over all functionality of the OS, and turned the whole computer into a browser appliance!

This complexity creates a vast attack surface, in need of regular patching. It's so complex that the whole apparatus is *de facto* fundamentally insecure – why else the need for continual patching? You can't stay safe without accepting updates, and the updates come with feature churn mixed in alongside the security patches in a mandatory undifferentiated trashstream of updates. (There is an Extended Support Release of Firefox which receives security patches for as long as a *whole year!*).

You might wonder what the problem is with all these specs growing and browsers getting regular updates. Isn't it a good thing that we enjoy rapid progress?

To the extent that we get to enjoy things like YouTube and sandspiel, yes! But to the extent that we want the internet to be a place where we can work and live and think and communicate free of malware, surveillance, dark patterns and the insidious influence of advertising, the answer is, empirically, sadly, no. The web has become ad-corrupted hand-in-hand with growth in technological capability, and the symbiotic relationship between *web* and *browser* means they feed on each others' churn. Ads demand new sources of novelty to put themselves on, so the web expands continually, the specs grow in complexity, the browsers grow in sophistication, the barrier to entry grows ever higher, the vast cost of it all demands more ad revenue to fund it… and thus the perpetual motion machine is complete.

There is no baseline, and no stability or clear standard or objective beyond continual expansion. **But stable standards are incredibly important**. They allow software, at least in theory, to be *finished.* Why is it important that software be finished? Because it gives us hope that we might *end the churn* and *fix all the bugs*! I want to use software whose version number is

1.0. I want to use software whose every line of code has been studied, analysed, optimised and punishingly tested. I want every component and subcomponent and every interaction and every configuration to be exquisitely documented, and taught in courses, and painstakingly deconstructed and proven sound. I want an ethnography of computer science whose scripture is this divine code, quoted for wisdom, on which scholars publish volumes of commentaries, and commentaries on the commentaries. I want software that *works*!

I can dream, right?

The neverending[6] churn as the web increases its scope *directly drives* the "Black hole of fad frameworks and brittle dependencies". In return for the marvel that is the modern web application, we have struck an awful bargain: it is now almost impossible to read a simple document on the web without signing up to the whole web application contraption, and there is no simple document-oriented subset of HTML for which you could create a simple document-oriented browser. Moreover, the web is now *everything* (witness the death of the thick client).

I posit no conspiracy theory here. It's the Systems Razor: Never attribute to malice or stupidity that which is adequately explained by uncontrolled complexity.

Can we fix it?

~~It's time to rebuild the web! Let's define a simplified subset of HTML, CSS and JavaScript that cannot be subverted into AdTech, and a simple, secure browser to consume it. Content will be delivered over a new, modern decentralised protocol...~~

Haha, only kidding. I can't be the only one tired of experimental proto-projects which promise revolution if only literally everybody could be persuaded to join their new network and become software technicians.

The ad-browser-social-network-industrial-complex is all but extinguishing the old web, and the foundationless, ad-driven churn driving it is harming the global information ecology and ultimately harming public discourse by making the open web a toxic experience and driving users to centralised platforms. Chaos and banditry force the population into the arms of feudal lords who promise protection.

6 HTML is now a "living standard"

There used to be an *internet middle class*, of non-commercial users who were not overtly technical, but were still able to self-publish. Much as in the offline world, the middle class has been squeezed, and now the internet population seems sharply polarised, with elite wizard hackers running things like IPFS on one side, and captive non-techies never seeing anything outside the Facebook mobile app.

Ironically, it's never been easier to set up a website – we have Medium, Wordpress, Wix, Blogger, and ten thousand other services providing infrastructure. But they are mostly used for evil[7], being the tools behind the satanic cabal of recipe sites, listicle-pushers, and clickbait merchants. Many genuine bloggers are out there in the web countryside, but drowned out.

I sorely miss the independent online citizenry with publishing sovereignty, but I just don't trust the browser-oriented web any more. What should we do?

Backwards to the future

> *"In anything at all, perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away, when a <body> has been stripped down to its nakedness." - Antoine de Saint-Exupéry*

Let HTML5 become the web application platform. Let the browser vendors keep developing forever more. We can't fight them (and besides, we all still want to watch YouTube), but we *can* escape to the countryside and rebuild something human-scale, human-controlled and human-understandable!

It is true that the PDF spec has suffered feature creep (3D models?!), so **we should use PDF/A** instead, which forbids interactive content (normal PDFs can contain JavaScript!) and ensures your PDFs are absolutely self-contained, even embedding the fonts.

"But how can you just throw away all of the semantic qualities of HTML?!"

HTML's semantic capabilities were oversold. Tagged PDF is just as expressive for all practical purposes. Maybe HTML supports richer metadata, but metadata is crap. Stick the right keywords in a document and 99% of semantic use cases are met.

"But why not use a traditional subset of HTML and just avoid JavaScript?!"

HTML encourages document creep. Are you disciplined enough to stick to that subset? Are you sure you won't be tempted to embed an interactive tweet or some other faustian convenience? How will your users trust that

7 Counterpoint: Neocities is awesome!

your site's plain-HTML-look doesn't hide some malicious JavaScript tracking anyway?

PDF/A makes a statement against such shenanigans.

"But what about RSS syndication?!"

You can point to a PDF file in an RSS feed if you wish.

"But how can I automate updates to my site's look and feel?!"

Same as with HTML – generate your PDFs using scripting. But unless you're fixing usability issues, do your users a favour and don't bother – readers are interested in the content and truly don't care about the headers, footers, sidebar, and whatever.

"But PDFs look so ugly when reading on my phone!"

Eh, they look alright to me. I do acknowledge that fancy spidery tiny text in multi-column layouts is awkward and ugly to read on a small screen, but that's a layout decision that you can make, and word processors (or DTP software) make it a whole lot easier for lay persons to control layout to suit their audience.

"But it's just as easy to write self-contained HTML pages!"

Sure, but if you're going to hide CTF forensics challenges in your publication, a coverdisk allows you to do it *in style*!

"But how can I implement shiny whizz-bang features that will engage readers and drive conversions?!"

You can't. PDF is boring. Boring is bad for business. But boring is good for users.

"But how can I add legitimate interactive features to my site, like user comments?!"

Foregoing interactive features is most certainly a sacrifice. But how much of a compromise is it *really* that commentators must get their own blog instead of commenting on yours? The instant gratification of in-line real-time comments is a double-edged sword. In my view, the great majority of comments are worthless, low-effort, trolling, psy-ops, or malicious. Comments worth making can withstand the friction of having to send an email. Raising the barrier to commentary could do the web some good. Not everything has to be scaled and automated.

I look forward to a new web niche of non-interactive yet richly-linked PDFs, each with a unique style and charm, born of innocent and uncomplicated

human desire to be heard, real monkeysphere-scale community, and freed from the dopamine treadmill of checking your likes, comments and engagement metrics.

"What are the thousands of browser engineers going to do all day now that you've destroyed HTML?"

Web applications aren't going anywhere any time soon, but perhaps some of that talent could be put to improving the PDF tool ecosystem. Despite the support I'm expressing here, I do acknowledge that PDF tools have a *lot* of rough edges and there is much room for improvement (Libreoffice is currently state of the art for open source tools that can generate Tagged PDF!).

PDF: Still Unfit for Human Consumption, 20 Years Later is a recent roast of PDF, but there are many fine counterpoints and rebuttals on Hacker News. I get it, PDF can be ugly. I'm still saying it's worth it to escape the noise.

"Come on, who are you kidding with all that moaning about infinite scrolling and advertising? If you don't like infinite scrolling, HTML doesn't force you to do it! Same goes for advertising."

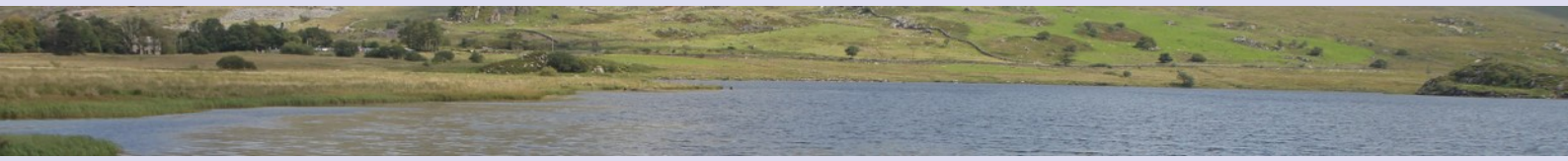
Sure, you can write good HTML. I won't argue with that. And if you're writing good HTML, good for you. But HTML is a dual-use technology, the bad guys are dual-using it an awful lot, and I feel that the stone age still has a part to play in the progression of the information age.

Call to action

Publish in static file formats.

Date and hash your work.

Stop spying on your users



Colophon

Taking my own advice, this document was written in the world's greatest web authoring tool: LibreOffice Writer. It was exported to Tagged PDF/A-2b, followed by adding the coverdisk as a PDF attachment, followed by qpdf and sed munging to achieve PDF/A-3b compliance. The coverdisk is a 1.44MB FAT-12 formatted floppy disk image containing a GPG public key which may be used in future crypto ops.

The document's primary URL is https://lab6.com/0 – although mirrors are welcome.

The document may contain outrageous falsehoods and embarrassing mistakes, unretractable due to it being immutable. Corrections may be published in future issues.

Contributions may be sent to the email address on the last page, or the Bitcoin address on the first page.

This document's hexadecimal SHA256 hash begins 0x00, identifying the issue number.

250e3f7d581acff115537ba38e89ad31 is a handy random 128-bit integer that will appear in every issue of Lab 6 and can be used to search for copies.

Tracking

OEIS entries	339,866
English Wikipedia entries	6,221,353
RC5-72 keys tested (% complete)	345,522,707,151,022,194,688 (7.317%)
Bitcoin blocks (hash)	664216 000000000000000000000348213d863624c61f0e8a2b1087236e0676d9046ff7a1
XKCDs	2406
IPv6 adoption	34.16%
Abe Vigoda Status Page Status	Still up
Largest Prime	$2^{82,589,933} - 1$
BTTF Movies	3
World population according to the CIA	7,684,292,383
Latest stable Linux kernel	5.10.4
Color Names	2,146,319

Hazcard Orange
#FF9906

Did you enjoy these numbers? Send your feedback to comment@input.lab6.com and it may be published in the next issue.