This is the text version of a talk I gave on October 29, 2015, at the Web Directions conference in Sydney. [53 minute video].

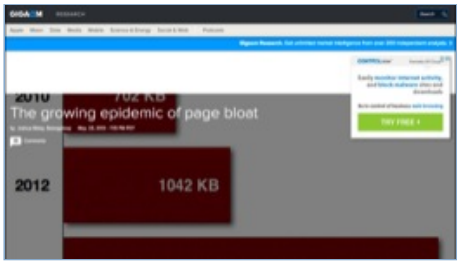
The Website Obesity Crisis

Let me start by saying that beautiful websites come in all sizes and page weights. I love big websites packed with images. I love high-resolution video. I love sprawling Javascript experiments or well-designed web apps.

This talk isn't about any of those. It's about mostly-text sites that, for unfathomable reasons, are growing bigger with every passing year.

While I'll be using examples to keep the talk from getting too abstract, I'm not here to shame anyone, except some companies (Medium) that should know better and are intentionally breaking the web.

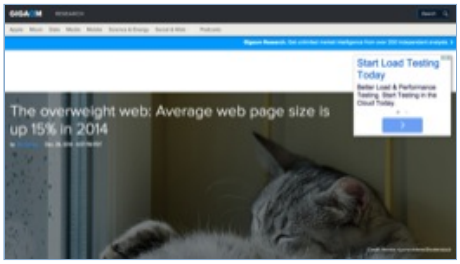
The Crisis



What do I mean by a website obesity crisis?

Here's an article on GigaOm from 2012 titled "The Growing Epidemic of Page Bloat". It warns that the average web page is over a megabyte in size.

The article itself is 1.8 megabytes long.



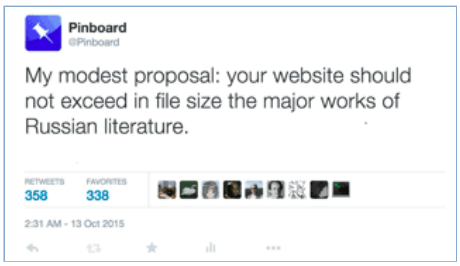
Here's an almost identical article from the same website two years later, called "The Overweight Web". This article warns that average page size is approaching 2 megabytes.

That article is 3 megabytes long.

If present trends continue, there is the real chance that articles warning about page bloat could exceed 5 megabytes in size by 2020.

The problem with picking any particular size as a threshold is that it encourages us to define deviancy down. Today's egregiously bloated site becomes tomorrow's typical page, and next year's elegantly slim design.

I would like to anchor the discussion in something more timeless.



To repeat a suggestion I made on Twitter, I contend that **text-based websites should not exceed in size the major works of Russian literature**.

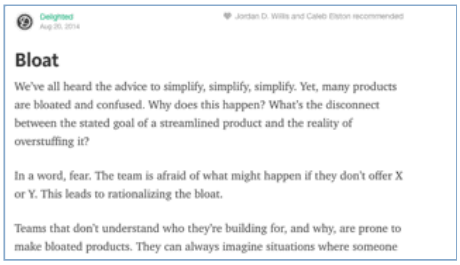
This is a generous yardstick. I could have picked French literature, full of slim little books, but I intentionally went with Russian novels and their reputation for ponderousness.

In Goncharov's Oblomov, for example, the title character spends the first hundred pages just getting out of bed.

If you open that tweet in a browser, you'll see the page is 900 KB big.

That's almost 100 KB more than the full text of The Master and Margarita, Bulgakov's funny and enigmatic novel about the Devil visiting Moscow with his retinue (complete with a giant cat!) during the Great Purge of 1937, intercut with an odd vision of the life of Pontius Pilate, Jesus Christ, and the devoted but unreliable apostle Matthew.

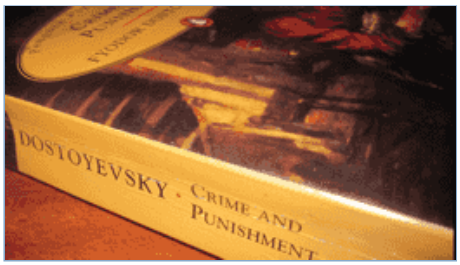
For a single tweet.



Or consider this 400-word-long Medium article on bloat, which includes the sentence:

"Teams that don't understand who they're building for, and why, are prone to make bloated products."

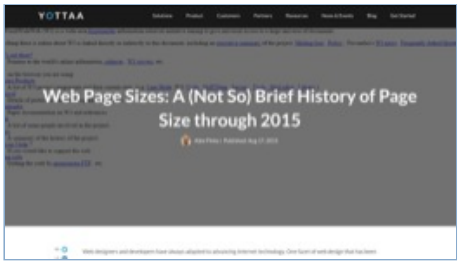
The Medium team has somehow made this nugget of thought require 1.2 megabytes.



That's longer than Crime and Punishment, Dostoyevsky's psychological thriller about an impoverished student who fills his head with thoughts of Napoleon and talks himself into murdering an elderly money lender.

Racked by guilt, so rattled by his crime that he even forgets to grab the money, Raskolnikov finds himself pursued in a cat-and-mouse game by a clever prosecutor and finds redemption in the unlikely love of a saintly prostitute.

Dostoevski wrote this all by hand, by candlelight, with a goddamned feather.



Here's a recent article called "A (Not So) Brief History of Page Bloat.

Rehearsing the usual reasons why bloat is bad, it includes the sentence "heavy pages tend to be slow pages, and slow pages mean unhappy users."

That sentence might put you in mind of the famous opening line to Anna Karenina:

"All happy families are alike; every unhappy family is unhappy in its own way."

But the not-so-brief history of bloat is much longer than Anna Karenina.

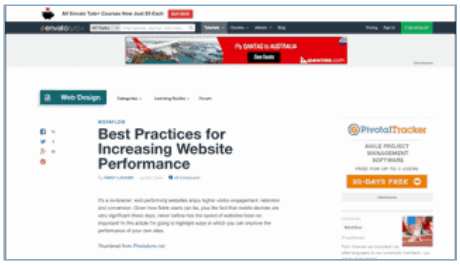
In fact, it's longer than War and Peace, Tolstoi's exploration of whether individual men and women can be said to determine the great events of history, or whether we are simply swept along by an irresistible current of historical inevitability.

Here's an article from the Yorkshire Evening Post, typical of thousands of local news sites. It does not explore the relationship between history and individual will at all:

"Leeds Hospital Bosses Apologise After Curry and Crumble On The Same Plate".

This poignant story of two foods touching on a hospital plate could almost have been written by Marcel Proust, for whom the act of dipping a morsel of cake in a cup of tea was the starting point for an expanding spiral of vivid recollections, culminating in the realization, nine volumes and 3 megabytes of handwritten prose later, that time and memory themselves are only an illusion.

The javascript alone in "Leeds Hospital Bosses Apologise after Curry and Crumble On The Same Plate" is longer than *Remembrance of Things Past*.



I could go on in this vein. And I will, because it's fun!

Here is an instructional article on Best Practices for Increasing Online performance that is 3.1 MB long.

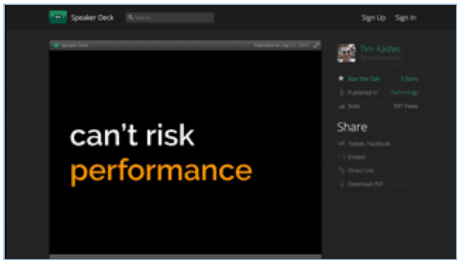
The article mentions that Google was able to boost user engagement in Google Maps by reducing the page weight from 100KB to 80KB.

Remember when Google Maps, the most sophisticated web app of its day, was thirty-five times smaller than a modern news article?

Web obesity can strike in the most surprising places.

Tim Kadlec, for example, is an excellent writer on the topic of performance. His personal site is a model of parsimony. He is full of wisdom on the topic of reducing bloat.

But the slides from his recent talk on performance are only available as a 9 megabyte web page, or a 14 megabyte PDF.



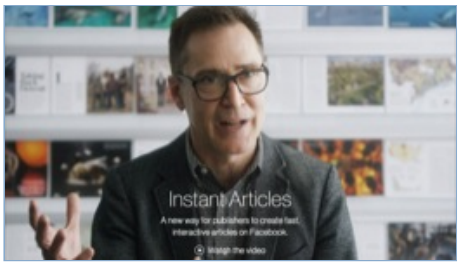
Let me close with a lovely TechTimes article warning that Google is going to start labeling huge pages with a special 'slow' mark in its mobile search interface.

The article somehow contrives to be 18 megabytes long, including (in the page view I measured) a 3 megabyte video for K-Y jelly, an "intimate lubricant".

It takes a lot of intimate lubricant to surf the unfiltered Web these days.

What the hell is up?

Fake Fixes



Everyone admits there's a problem. These pages are bad enough on a laptop (my fan spun for the entire three weeks I was preparing this talk), but they are hell on mobile devices. So publishers are taking action.

In May 2015, Facebook introduced 'Instant Articles', a special format for news stories designed to appear within the Facebook site, and to load nearly instantly.

Facebook made the announcement on a 6.8 megabyte webpage dominated by a giant headshot of some dude. He doesn't even work for Facebook, he's just the National Geographic photo editor.

Further down the page, you'll find a 41 megabyte video, the only way to find out more about the project. In the video, this editor rhapsodizes about exciting misfeatures of the new instant format like tilt-to-pan images, which means if you don't hold your phone steady, the photos will drift around like a Ken Burns documentary.

Facebook has also launched internet.org, an effort to expand Internet access. The stirring homepage includes stories of people from across the developing world, and what getting Internet access has meant for them.

You know what's coming next. When I left the internet.org homepage open in Chrome over lunch, I came back to find it had transferred over a quarter gigabyte of data.

Surely, you'll say, there's no way the globe in the background of a page about providing universal web access could be a giant video file?

But I am here to tell you, oh yes it is. They load a huge movie just so the globe can spin.

This is Facebook's message to the world: "The internet is slow. Sit and spin."

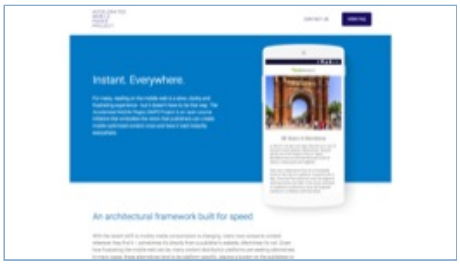
And it's not like bad connectivity is a problem unique to the Third World! I've traveled enough here in Australia to know that in rural places in Tasmania and Queensland, vendors treat WiFi like hundred-year-old brandy.

You're welcome to buy as much of it as you want, but it costs a fortune and comes in tiny portions. And after the third or fourth purchase, people start to look at you funny.

Even in well-connected places like Sydney, we've all had the experience of having a poor connection, and almost no battery, while waiting for some huge production of a site to load so we can extract a morsel of information like a restaurant address.

The designers of pointless wank like that Facebook page deserve the ultimate penalty.

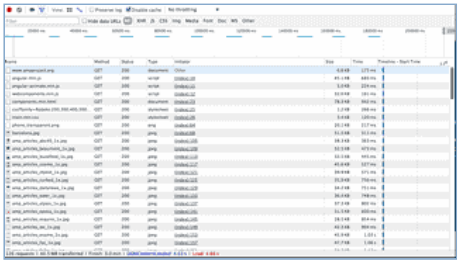
They should be forced to use the Apple hockey puck mouse for the remainder of their professional lives. [shouts of horror from the audience]



Google has rolled out a competitor to Instant Articles, which it calls Accelerated Mobile Pages. AMP is a special subset of HTML designed to be fast on mobile devices.

Why not just serve regular HTML without stuffing it full of useless crap? The question is left unanswered.

The AMP project is ostentatiously open source, and all kinds of publishers have signed on. Out of an abundance of love for the mobile web, Google has volunteered to run the infrastructure, especially the user tracking parts of it.



Jeremy Keith pointed out to me that the page describing AMP is technically infinite in size. If you open it in Chrome, it will keep downloading the same 3.4 megabyte carousel video forever.

If you open it in Safari, where the carousel is broken, the page still manages to fill 4 megabytes.

These comically huge homepages for projects designed to make the web faster are the equivalent of watching a fitness video where the presenter is just standing there, eating pizza and cookies.

The world's greatest tech companies can't even make these tiny text sites, describing their flagship projects to reduce page bloat, lightweight and fast on mobile.

I can't think of a more complete admission of defeat.

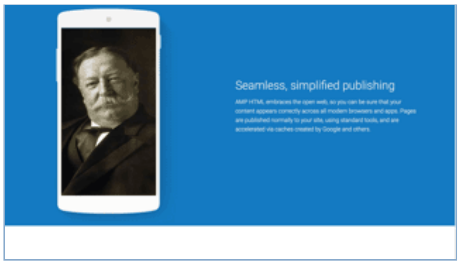
The tech lead for Google's AMP project was nice enough to engage us on Twitter. He acknowledged the bloat, but explained that Google was "resource constrained" and had had to outsource this project.

This admission moved me deeply, because I had no idea Google was in

a tight spot. So I spent a couple of hours of my own time making a static version of the AMP website.

I began by replacing the image carousels with pictures of William Howard Taft, America's greatest president by volume.

I think this made a marked improvement from the gratuitous animations on the original page.



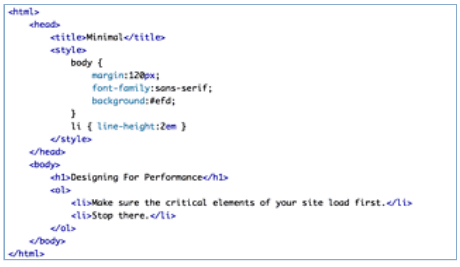
By cutting out cruft, I was able to get the page weight down to half a megabyte in one afternoon of work. This is eight times smaller than the original page.

I offered my changes to Google free of charge, but they are evidently too resource constrained to even find the time to copy it over.

This project led me to propose the **Taft Test**:

Does your page design improve when you replace every image with William Howard Taft?

If so, then, maybe all those images aren't adding a lot to your article. At the very least, leave Taft there! You just admitted it looks better.



```
<html>
    <head>
        <title>Minimal</title>
        <style>
            body {
                margin:120px;
                font-family:sans-serif;
                background:#efd;
            }
            li { line-height:2em }
        </style>
    </head>
    <body>
        <h1>Designing For Performance</h1>
        <ol>
            <li>Make sure the critical elements of your site load first.</li>
            <li>Stop there.</li>
        </ol>
    </body>
</html>
```

I want to share with you my simple two-step secret to improving the performance of any website.

1. Make sure that the most important elements of the page download and render first.

2. Stop there.

You don't need all that other crap. Have courage in your minimalism.

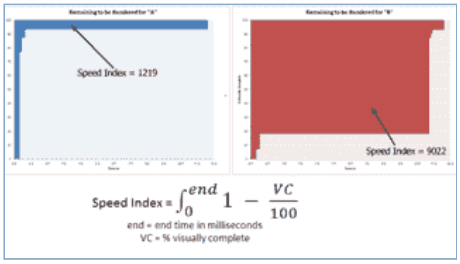
To channel a famous motivational speaker, I could go out there tonight, with the materials you've got, and rewrite the sites I showed you at the start of this talk to make them load in under a second. In two hours.

Can you? Can you?

Of course you can! It's not hard! We knew how to make small websites in 2002. It's not like the secret has been lost to history, like Greek fire or Damascus steel.

But we face pressure to make these sites bloated.

I bet if you went to a client and presented a 200 kilobyte site template, you'd be fired. Even if it looked great and somehow included all the tracking and ads and social media crap they insisted on putting in. It's just so far out of the realm of the imaginable at this point.



If you've ever struggled to lose weight, you know there are tricks people use to fool themselves into thinking they're thinner. You suck in your gut, wear a tight shirt, stand on a certain part of the scale.

The same situation obtains with performance testing. People have invented creative metrics to persuade themselves that their molasses-like websites load fast.

Google has a popular one called SpeedIndex. (You know it's from Google because they casually throw an integral sign into the definition.)

SpeedIndex is based on the idea that what counts is how fast the visible part of the website renders. It doesn't matter what's happening elsewhere on the page. It doesn't matter if the network is saturated and your phone is hot to the touch. It doesn't matter if the battery is visibly draining. Everything is OK as long as the part of the site in the viewport appears to pop into view right away.

Of course, it doesn't matter how fast the site appears to load if the first thing the completed page does is serve an interstitial ad. Or, if like many mobile users, you start scrolling immediately and catch the 'unoptimized' part of the page with its pants down.

There is only one honest measure of web performance: **the time from when you click a link to when you've finished skipping the last ad**.

Everything else is bullshit.

In conversations with web performance advocates, I sometimes feel like a hippie talking to SUV owners about fuel economy.

They have all kinds of weirdly specific tricks to improve mileage. Deflate the front left tire a little bit. Put a magnet on the gas cap. Fold in the side mirrors.

Most of the talk about web performance is similarly technical, involving compression, asynchronous loading, sequencing assets, batching HTTP requests, pipelining, and minification.

All of it obscures a simpler solution.

If you're only going to the corner store, ride a bicycle.

If you're only displaying five sentences of text, use vanilla HTML. Hell, serve a textfile! Then you won't need compression hacks, integral signs, or elaborate Gantt charts of what assets load in what order.

Browsers are really, really good at rendering vanilla HTML.

We have the technology.

Nutritionists used to be big on this concept of a food pyramid. I think we need one for the web, to remind ourselves of what a healthy site should look like.

Here is what I recommend for a balanced website in 2015:

- A solid base of text worth reading, formatted with a healthy dose of markup.
- Some images, in moderation, to illustrate and punch up the visual design.
- A dollop of CSS.
- And then, very sparingly and only if you need it, JavaScript.

Instead, here is the web pyramid as we observe it in the wild:

- A base layer of HTML
- A huge pile of crap
- On top of it all, a whole mess of surveillance scripts.

Fat Ads

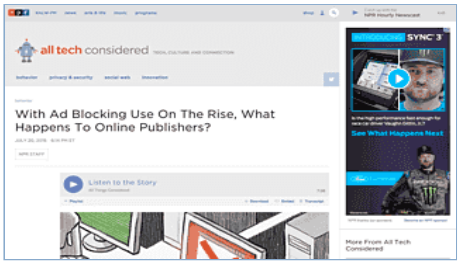
Web designers! It's not all your fault.

You work your heart out to create a nice site, optimized for performance. You spend the design process trying to anticipate the user's needs and line their path with rose petals.

Then, after all this work is done, your client makes you shit all over your hard work by adding tracking scripts and ads that you have no control over, whose origin and content will be decided at the moment the page loads in the user's browser, and whose entire purpose is to break your design and distract the user from whatever they came to the site to do.

The user's experience of your site is dominated by hostile elements out of your control.

This is a screenshot from an NPR article discussing the rising use of ad



blockers. The page is 12 megabytes in size in a stock web browser.

The same article with basic ad blocking turned on is one megabyte. It's no model of parsimony, but still, what a difference a plugin makes.

If you look at what the unblocked version pulls in, it's not just videos and banner ads, but file after file of javascript. Every beacon, tracker and sharing button has its own collection of scripts that it needs to fetch from a third-party server. Each request comes packed with cookies.

More cookies are the last thing your overweight website needs.

These scripts get served from God knows where and are the perfect vector for malware.

Advertisers will tell you it has to be this way, but in dealing with advertisers you must remember they are professional liars.

I don't mean this to offend. I mean it as a job description. An advertiser's job is to convince you to do stuff you would not otherwise do. Their task in talking to web designers is to persuade them that the only way to show ads is by including mountains of third-party cruft and tracking.

The bloat, performance, and security awfulness, they argue, is the price readers pay for free content.

I've come across these diagrams of the "adtech ecosystem", which I love. They communicate the sordidness of advertising in the way simple numbers never could.

Here is a view of the adtech ecosystem in 2011, when there were 100 'adtech' companies.

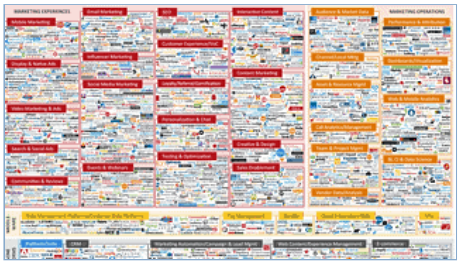
Here's how things stood in 2012, when there were 350 of them.

By 2014, we were blessed with 947.

And in 2015 we have 1876 of these things. They are all competing for the same little slice of your online spending.

This booming industry is very complex—I believe intentionally so.

When you're trying to understand a complex system, it can be helpful to

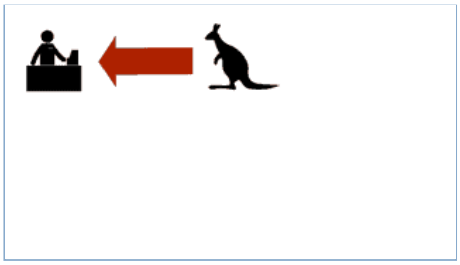


zoom out and look at the overall flow of things.

For example, here's a German diagram showing the energy budget of the Earth.

All kinds of complicated things happen to sunlight when it shines on plants or water, but you can ignore them completely and just measure the total energy that comes in and out.

In the same spirit, let me sketch the way money is flowing in to the advertising bubble.



In the beginning, you have the consumer. In a misguided attempt at cultural sensitivity, I have chosen to represent the consumer with a kangaroo.

Consumers give money to merchants in exchange for goods and services. Here the red arrow represents money flowing to the merchant, or as you say in Australia, "dollars".

A portion of this money is diverted to pay for ads. Think of it as a little consumption tax on everything you buy.

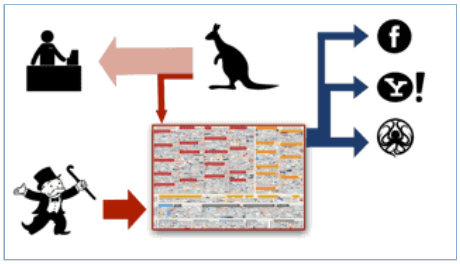
This money bounces around in the world of advertising middlemen until it ultimately flows out somewhere into someone's pocket.

Right now it's ending up in the pockets of successful ad network operators like Facebook, Yahoo!, and Google.

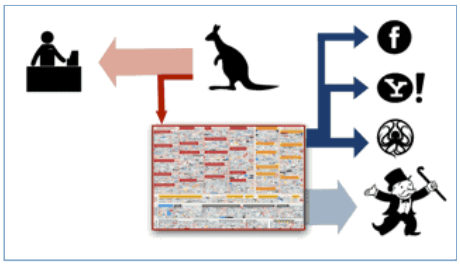
You'll notice that there's more money flowing out of this system than into it.

There's a limit to how much money is available to ad companies from just consumers. Think of how many ads you are shown in a given day, compared to the number of purchases you actually make.

So thank God for investors! Right now they are filling the gap by pouring funding into this white-hot market. Their hope is that they will



pick one of the few companies that ends up a winner.



However, at some point the investors who are pouring money in will want to move to the right-hand side of this diagram. And they'll want to get back even more money than they invested.

When this happens, and I believe it is happening right now, something will have to give.

Either we start buying more stuff, or a much bigger portion of our purchases goes to pay for ads…

Or the bubble is going to burst.

As it bursts, the remaining ad startups will grow desperate. They will search for ways to distinguish themselves from the pack with innovative forms of surveillance.

We'll see a wave of consolidation, mergers, aggressive new forms of tracking, and the complete destruction of what remains of online privacy.

This why I've proposed we regulate the hell out of them now.

I think we need to ban third-party tracking, and third party ad targeting.

Ads would become dumb again, and be served from the website they appear on.

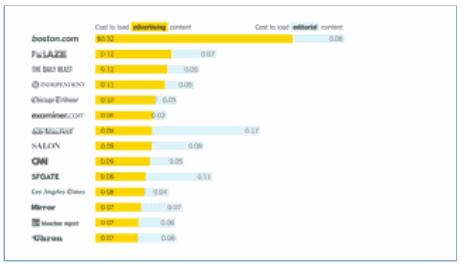
Accepted practice today is for ad space to be auctioned at page load time. The actual ads (along with all their javascript surveillance infrastructure) are pulled in by the browser after the content elements are in place.

In terms of user experience, this is like a salesman arriving at a party after it has already started, demanding that the music be turned off, and setting up their little Tupperware table stand to harass your guests. It ruins the vibe.

Imagine what server-side ad layout would mean for designers. You would actually know what your pages are going to look like. You could load assets in a sane order. You could avoid serving random malware.

Giant animations would no longer helicopter in at page load time,

destroying your layout and making your users hate you.



In fact, let's be even bolder in our thinking. I'm not convinced that online publishing needs to be ad-supported at all.

People dismiss micropayments, ignoring the fact that we already have a *de facto* system of micropayments that is working well.

This chart from the New York Times shows how much money you spend per page load on an American cell phone network, based on the bandwidth used. For example, it costs thirty cents to load a page from Boston.com on a typical data plan.

This is nothing more than a micropayment to the telecommunications company. And I'm sure it's more revenue than Boston.com sees from the ad impressions on the page.

We're in a stupid situation where ads make huge profits for data carriers and ad networks, at the expense of everyone else.

Advertisers will kick and scream at any attempt to make them go back to the dumb advertising model. But there's no evidence that dumb ads are any worse than smart ones.

For years and years, poorly targeted advertising brought in enough money to fund entire television studios, radio shows, and all kinds of popular entertainment. Dumb ads paid for the Batmobile.

It costs a lot less to pay for a couple freelance journalists and a web designer than it does to film a sitcom. So why is it unthinkable to force everyone back to a successful funding model that doesn't break privacy?

Of course, advertisers will tell us how much better TV in the old days could have been if they had been able to mount a camera on top of every set.

But we've heard enough out of them.

Dumb ads will mean less ad revenue, because a lot of online ad spending is fueled by extravagant promises around the possibilities of surveillance technology.

But the ad market is going to implode anyway when the current bubble bursts.

The only question for publishers is whether to get ahead of this and reap the benefits, or circle down the drain with everybody else.

Fat Assets

Let’s talk about a different cause of web obesity.

Fat assets!

This has been a problem since forever, but as networks get faster, and publishing workflows get more complicated, it gets easier to accidentally post immense files to your website.

Examples!

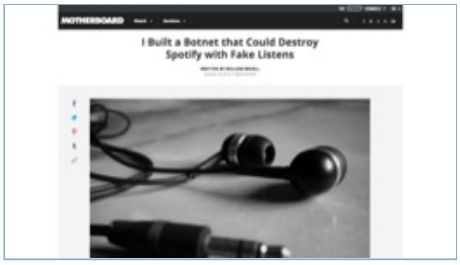
Here’s a self-righteous blogger who likes to criticize others for having bloated websites. And yet there's a gratuitous 3 megabyte image at the top of his most recent post.

Presumably this was a simple case of forgetting to resize an image. Without loading it on a slow connection, it's hard to notice the mistake.

Making networks faster makes this problem worse.

Here's a recent photo of a traffic jam in China. There are 50 lanes of cars here. Adding a 51st lane is not going to make things any better.

Similarly, adding network capacity is not going to convince people to start putting less stuff on their website.



Consider this recent Vice article about botnets.

At the top of the article is a pointless 3 megabyte photograph of headphones. This page fails the Taft Test.

This is part of a regrettable trend, made possible by faster networks, of having ‘hero images’ whose only purpose is for people to have something to scroll past.

In this case there's no use blaming the author. Something in the publishing toolchain failed to minimize this enormous image.

But the larger problem is that fast networks encourage people to include this kind of visual filler.

As we rely more and more on compression tricks, minimization, caching, and server configuration, mistakes become harder to catch and potentially more expensive.

Here's another example, interesting for two reasons.

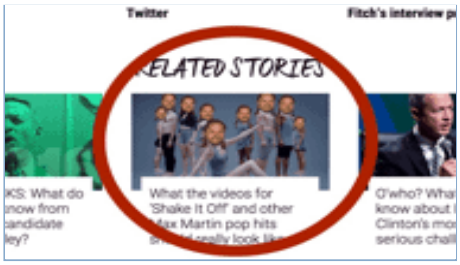
First, the original image quality is awful. The picture looks like it was

taken with a potato because it's a screen capture from a TV show.

Nevertheless, the image is enormous. If you load this website in Safari, the image is several megabytes in size.

If you load it in Chrome, it's 100 kilobytes, because Chrome supports an on-the-fly compression format that Safari doesn't.

With these complicated optimization pipelines, it's hard to be sure you're seeing the same thing as your audience.



As a bonus, if you scroll to the bottom of the page, you see that a tiny animated GIF in the part of the page layout designers call "chum" is over a megabyte in size.

It's an useless piece of clickbait, but contributes massively to the overall weight of the page.

No one has fatter assets than Apple. Their site is laughably bloated. I think it may be a marketing trick.

"These images load crazy slow on my crappy Android phone, I can't wait to get one of those Apple devices!"

Let's take a look at the Apple page that explains iOS on the iPad Pro.

How big do you think this page is?

Would you believe that it's bigger than the entire memory capacity of the iconic iMac? (32 MB)

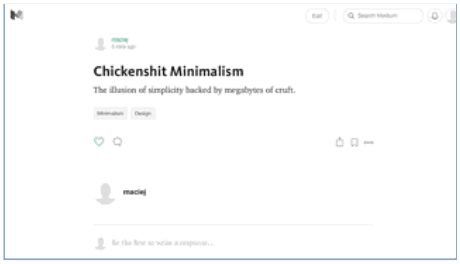
In fact, you could also fit the contents of the Space Shuttle Main Computer. Not just for one Shuttle, but the entire fleet (5 MB).

And you would still have room for a tricked out Macintosh SE... (5MB).

...and the collected works of Shakespeare... (5 MB)

With lots of room to spare. The page is 51 megabytes big.

Chickenshit Minimalism



These Apple sites exemplify what I call Chickenshit Minimalism. It's the prevailing design aesthetic of today's web.

I wrote an essay about this on Medium. Since this is a fifty minute talk, please indulge me while I read it to you in its entirety:

"Chickenshit Minimalism: the illusion of simplicity backed by megabytes of cruft."

I already talked about how bloated Medium articles are. That one-sentence essay is easily over a megabyte.

It's not just because of (pointless) javascript. There's also this big image in the page footer.

Because my article is so short, it's literally impossible to scroll down to see it, but with developer tools I can kind of make out what it is: some sort of spacesuit people with tablets and mobile phones.

It's 900 kilobytes in size.

Here's another example of chickenshit minimialism: the homepage for Google's contributor program.

This is a vast blue wasteland, 2 megabytes in size, that requires you to click three times in order to read three sentences.

The last sentence will tell you that the program is not available here in Australia.

Here's the homepage for the Tatamagouche Brewing Company. The only thing on it is a delicious beer. All the navigation has been tucked away into a hamburger menu.

Tucking into hamburgers is not the way to fix your flabby interface.

Design companies love this invisible hamburger antipattern.

Here's the 3 megabyte homepage for a company called POLLEN. You can barely even see the hamburger up there.

For the ultimate example of the chickenshit aesthetic, take a look at Verge's review of the Apple watch.

But please don't load this on your phones right now, or you're going to bring down the conference wifi.

The Verge review is a UI abomination that completely hijacks the scroll mechanic of your browser. As you try to scroll down, weird things happen.

Interface elements slide in from the left.

Interface elements slide in from the right.

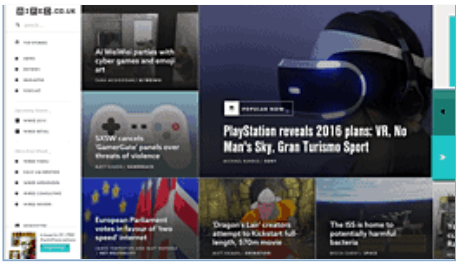
Interface elements you haven't seen since middle school call you unexpectedly in the middle of the night.

Once in a great while, the page actually scrolls down.

And what mainly happens is the fan on your laptop spins for dear life.

I tried to capture a movie of myself scrolling through the Verge Apple watch review, but failed. The graphics card on my late-model Apple laptop could literally not cope with the load.

Interface Sprawl

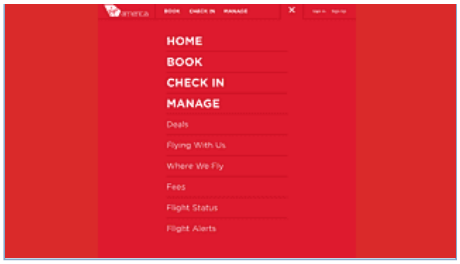


Some kind of brain parasite infected designers back when the iPad came out, and they haven't recovered. Everything now has to look like a touchscreen.

This is the UK version of Wired, another site that has declared war on the scroll event.

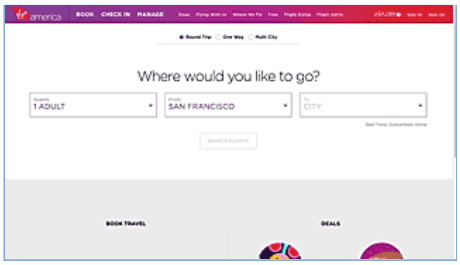
You can try to scroll down, but it will just obstinately move you to the right instead. Article titles show up as giant screen-eating tiles of cruft.

Another hallmark of iPad chic are these elegant infographics in unreadable skinny white font on a light background.



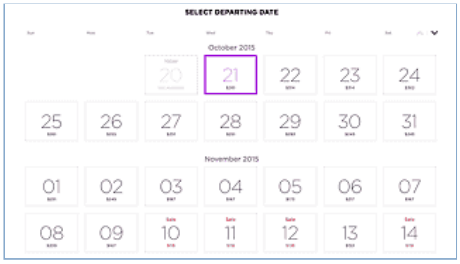
Book a flight on Virgin America and you'll encounter this column of giant buttons floating in a sea of red.

This interface may look clean on a phone, but on a large screen it's just terrifying.



The "Book" button on that screen takes you to a land of vast input fields.

Note the hallmark ecosystem of giant fonts, tiny fonts, and extremely pale fonts.

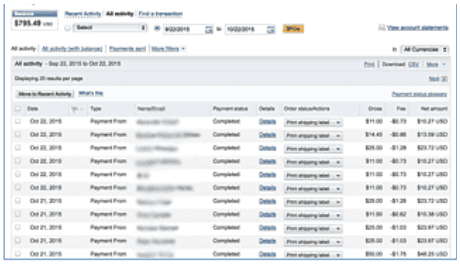


After you decide where to go, the site takes you to this calendar widget.

It has equally enormous buttons, but the only piece of information I'm interested in—the price of the flight on each day—appears in microscopic type under the date.

My gripe with this design aesthetic is the loss of information density. I'm an adult human being sitting at a large display, with a mouse and keyboard. I deserve better.

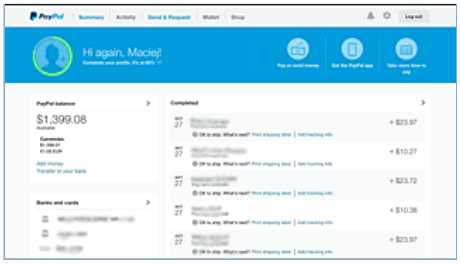
Not every interface should be designed for someone surfing the web from their toilet.



Here's what the PayPal site used to look like.

I never fell to my knees to thank God for giving me the gift of sight so that I might behold the beauty of the old PayPal interface.

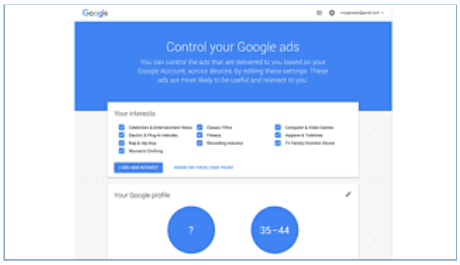
But it got the job done.



Here's the PayPal website as it looks today.

The biggest element on the page is an icon chastising me that I haven't told PayPal what I look like. Next to that is a useless offer to 'download the app', and then an offer for a credit card.

I can no longer control the sort order, there are no filter tools, and you see there are far fewer entries visible without scrolling.

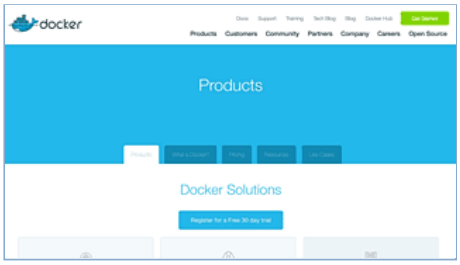


Here is a Google 'control panel' that lets you configure your 'ad preferences'. It is similarly toylike and visually bloated.

It's like we woke up one morning in 2008 to find that our Lego had all turned to Duplo. Sites that used to show useful data now look like cartoons.

Interface elements are big and chunky. Any hint of complexity has been pushed deep into some sub-hamburger.

Sites target novice users on touchscreens at everyone else's expense.

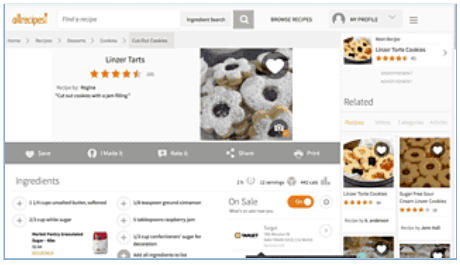


Another example of this interface bloat: the Docker homepage. It consists of faint text separated by enormous swathes of nothingness.

I shouldn't need sled dogs and pemmican to navigate your visual design.

Search pages are where the pain hits hardest. Giant lettering and fat buttons replace the one thing anyone needs to see—a list of search results.

Here's a design where there's room for only one result, again on a giant high-resolution monitor.



I hate to do it, but I have to call out responsive design.

Everyone recognizes that it's challenging to make a site that looks good at all screen sizes.

But the emphasis on screen size has obscured an important difference in how people interact with interface elements.

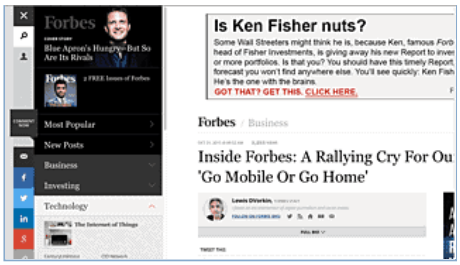
On a phone, people are poking at a small screen with the meat styluses hanging off their arms. In that scenario, it makes sense to have big buttons.

On a large screen, where you have acres of space and an exquisitely sensitive pointing device, the same interface is maddening.

There may be no way to split the difference. I feel like designers are just waiting for us all to stop using laptops.

This is a typical recipe site grappling with this UI problem. I don't want to pick on it, because it's trying very hard.

But notice how some elements are tiny, and some are huge. Half the page is in the idiom of touch interfaces, and the whole thing is hard to read.



Here is the Forbes homepage, as seen with the left hamburger menu expanded. It looks like a random chunk of memory that accidentally got rendered to the video card.

There are multiple icons for social sharing, up arrows, down arrows, a smorgasbord of fonts.

And sitting confidently atop it all is big a fat turd of a banner ad, with its own ideas about typography and layout.

This is no way to live. We're not animals!

Heavy Clouds

Finally, I want to talk about our giant backends.

How can we expect our web interfaces to be slim when we're setting such a bad example on the server side?

I have a friend who bakes cookies for a living. Like a lot of home bakers, she started by using her own kitchen, running it at full capacity until everything was covered in flour and her apartment was tropically hot.

At a certain point, she realized she needed to buy commercial baking equipment.

Being good at baking cookies doesn't teach you anything about how to buy professional restaurant equipment.

For a home cook, it's terrifying to have to purchase a commercial oven, cooling racks, an industrial mixer, and start buying ingredients in fifty-pound sacks.

It's even scarier to hire staff, rent kitchen space, and get health permits One mistake can end your business.

For years, the Internet worked the same way. You could run a small

website off of commodity servers, but if your project started gaining traction, you found yourself on the phone with a silken-voiced hardware salesperson about signing contracts for equipment, bandwidth, and colocation space.

It was easy to get out of your depth and make expensive mistakes.

Amazon Web Services changed everything. They offered professional tools on demand, by the hour, at scale. You didn't have to shop for hardware, and you weren't locked in to owning it. You paid a premium for the service, but it removed a ton of risk.

Of course, you still had to learn how to use this stuff. But that was actually fun.

There was always a catch. The gas burners on the stoves were kind of small. The handles would occasionally fall off the frying pans, at unexpected times.

And to their credit, Amazon warned you about this up front, and told you to design your procedures with failures in mind.

Some things were guaranteed to never fail—the freezers, say, would always stay below freezing. But maybe you wouldn't be able to unlock the doors for several hours at a time.

As people began moving to the cloud, it forced them to think at a bigger scale. They had to think in terms of multiple machines and availability zones, and that meant thinking about redundancy, failure tolerance.

All of these are good things at scale, but overkill for a lot of smaller sites. You don't need an entire restaurant kitchen staff to fry an egg.

As the systems got bigger, Amazon started offering more automation. They would not only rent you giant ovens, but a fleet of kitchen robots that you could program do all kinds of mundane tasks for you.

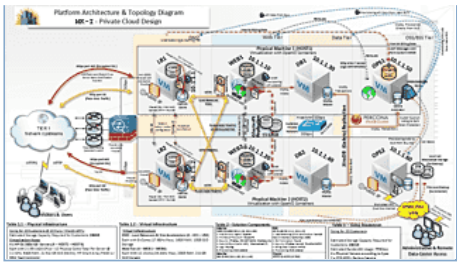
And again, it was way more fun to program the robots than to do the mundane kitchen tasks yourself.

For a lot of tech companies, where finding good programmers is harder than finding money, it made sense to switch over to the highly automated cloud services entirely.

For programmers, the cloud offered a chance to design distributed systems across dozens or hundreds of servers early in their careers. It was like getting the keys to a 747 right out of flight school.

Most website work is pretty routine. You hook up a database to a template, and make sure no one trips over the power cord.

But automation at scale? That's pretty sweet, and it's difficult!



It's like you took a bunch of small-business accountants and told them they were going to be designing multi-billion dollar corporate tax shelters in the Seychelles.

Suddenly they feel alive, they feel free. They're right at the top of Maslow's hierarchy of needs, self-actualizing on all cylinders. They don't want to go back.

That's what it feels like to be a programmer, lost in the cloud.

Complexity is like a bug light for smart people. We can't resist it, even though we know it's bad for us. This stuff is just so cool to work on.

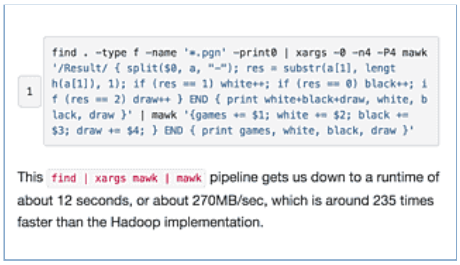
The upshot is, much of the web is horribly overbuilt.

Technologies for operating at scale developed by companies that need them end up in the hands of people who aspire to work at those scales.

And there's no one to say "no".

Adam Drake wrote an engaging blog post about analyzing 2 million chess games. Rather than using a Hadoop cluster, he just piped together some Unix utilities on a laptop, and got a 235-fold performance improvement over the 'Big Data' approach.

The point is not that people using Hadoop clusters are foolish, or that everything can be done on a laptop. It's that many people's intuition about what constitutes a large system does not reflect the reality of 2015 hardware.



You can do an awful lot on a laptop, or pizza box web server, if you skip the fifty layers of overhead.

	PINBOARD	ACME
DAILY USERS	8K	3K
EMPLOYEES	1/2	1/2
REVENUE	12K	5K
HOSTING	1K	9K
INCOME	11K	(4K)

Let me give you a concrete example. I recently heard from a competitor, let's call them ACME Bookmarking Co., who are looking to leave the bookmarking game and sell their website.

While ACME has much more traffic than I do, I learned they only have half the daily active users. This was reassuring, because the hard part of scaling a bookmarking site is dealing with people saving stuff.

We both had the same number of employees. They have an intern working on the project part time, while I dither around and travel the world giving talks. Say half a full-time employee for each of us.

We have similar revenue per active user. I gross $12,000 a month, they gross $5,000.

But where the projects differ radically is cost. ACME hosts their service on AWS, and at one point they were paying $23,000 (!!) in monthly fees. Through titanic effort, they have been able to reduce that to $9,000 a month.

I pay just over a thousand dollars a month for hosting, using my own equipment. That figure includes the amortized cost of my hardware, and sodas from the vending machine at the colo.

So while I consider bookmarking a profitable business, to them it's a $4,000/month money pit. I'm living large off the same income stream that is driving them to sell their user data to marketers and get the hell out of the game.

The point is that assumptions about complexity will anchor your expectations, and limit what you're willing to try. If you think a 'real' website has to live in the cloud and run across a dozen machines, a whole range of otherwise viable projects will seem unprofitable.

Similarly, if you think you need a many-layered CMS and extensive custom javascript for an online publishing venture, the range of things you will try becomes very constricted.

Rather than trying to make your overbuilt projects look simple, ask yourself if they can't just *be* simple.

I don't want to harsh on the cloud. Some of my best friends are in the cloud.

Rather, I want to remind everyone there's plenty of room at the bottom. Developers today work on top of too many layers to notice how powerful the technology has become. It's a reverse princess-and-the-pea problem.

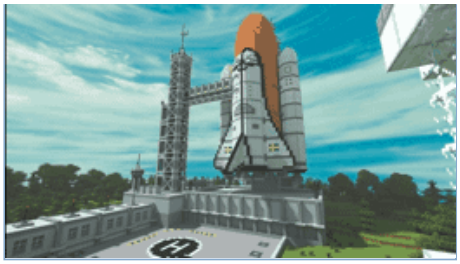
The same thing happens in the browser. The core technology is so fast and good that we've been able to pile crap on top of it and still have it work tolerably well.

One way to make your website shine is by having the courage to let the browser do what it's optimized to do. Don't assume that all your frameworks and tooling are saving you time or money.

Unfortunately, complexity has become a bit of a bragging point. People boast to one another about what's in their 'stack', and share tips about how to manage it.

"Stack" is the backend equivalent to the word "polyfill". Both of them are signs that you are radically overcomplicating your design.

Stirring Conclusion



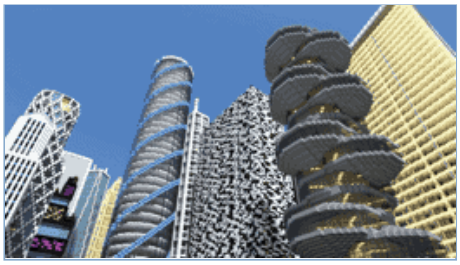
There's a reason to care about this beyond just aesthetics and efficiency.

Let me use a computer game analogy to express two visions of the future Web.

The first vision is the Web as Minecraft—an open world with simple pieces that obey simple rules. The graphics are kind of clunky, but that's not the point, and nobody cares.

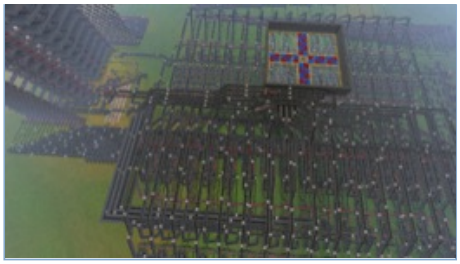
In this vision, you are meant to be an active participant, you're supposed to create stuff, and you'll have the most fun when you collaborate with others.

The rules of the game are simple and don't constrain you much.



People create astonishing stuff in Minecraft.

Here is an entire city full of skyscrapers, lovingly tended.



Here are some maniacs who have built an entire working CPU out of redstone. if this were scaled up big enough, it could also run Minecraft, which is a mind-bending thought.

The game is easy to learn and leaves you to your own devices. Its lack of polish is part of its appeal.

The other vision is of the web as Call of Duty—an exquisitely produced, kind-of-but-not-really-participatory guided experience with breathtaking effects and lots of opportunities to make in-game purchases.

Creating this kind of Web requires a large team of specialists. No one person can understand the whole pipeline, nor is anyone expected to. Even if someone could master all the technologies in play, the production costs would be prohibitive.

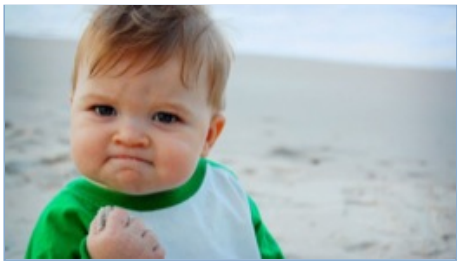
The user experience in this kind of Web is that of being carried along, with the illusion of agency, within fairly strict limits. There's an obvious path you're supposed to follow, and disincentives to keep you straying from it. As a bonus, the game encodes a whole problematic political agenda. The only way to reject it is not to play.

Despite the lavish production values, there's a strange sameness to everything. You're always in the same brown war zone.

With great effort and skill, you might be able make minor modifications to this game world. But most people will end up playing exactly the way the publishers intend. It's passive entertainment with occasional button-mashing.

Everything we do to make it harder to create a website or edit a web page, and harder to learn to code by viewing source, promotes that consumerist vision of the web.

Pretending that one needs a team of professionals to put simple articles online will become a self-fulfilling prophecy. Overcomplicating the web means lifting up the ladder that used to make it possible for people to teach themselves and surprise everyone with unexpected new ideas.



Here's the hortatory part of the talk:

Let’s preserve the web as the hypertext medium it is, the only thing of its kind in the world, and not turn it into another medium for consumption, like we have so many examples of already.

Let’s commit to the idea that as computers get faster, and as networks get faster, the web should also get faster.

Let’s not allow the panicked dinosaurs of online publishing to trample us as they stampede away from the meteor. Instead, let's hide in our holes and watch nature take its beautiful course.

Most importantly, let’s break the back of the online surveillance establishment that threatens not just our livelihood, but our liberty. Not only here in Australia, but in America, Europe, the UK—in every free country where the idea of permanent, total surveillance sounded like bad science fiction even ten years ago.

The way to keep giant companies from sterilizing the Internet is to make their sites irrelevant. If all the cool stuff happens elsewhere, people will follow. We did this with AOL and Prodigy, and we can do it again.

For this to happen, it's vital that the web stay participatory. That means not just making sites small enough so the whole world can visit them, but small enough so that people can learn to build their own, by example.

I don't care about bloat because it's inefficient. I care about it because it makes the web inaccessible.

Keeping the Web simple keeps it awesome.

Thank you very much!

HEAVY, ROILING, TROUBLED SEAS OF APPLAUSE

Sincere thanks to Michael Krakovskiy, Jeremy Keith, Nick Heer, and lots of Twitter friends for their help with this talk.

Idle Words > Talks