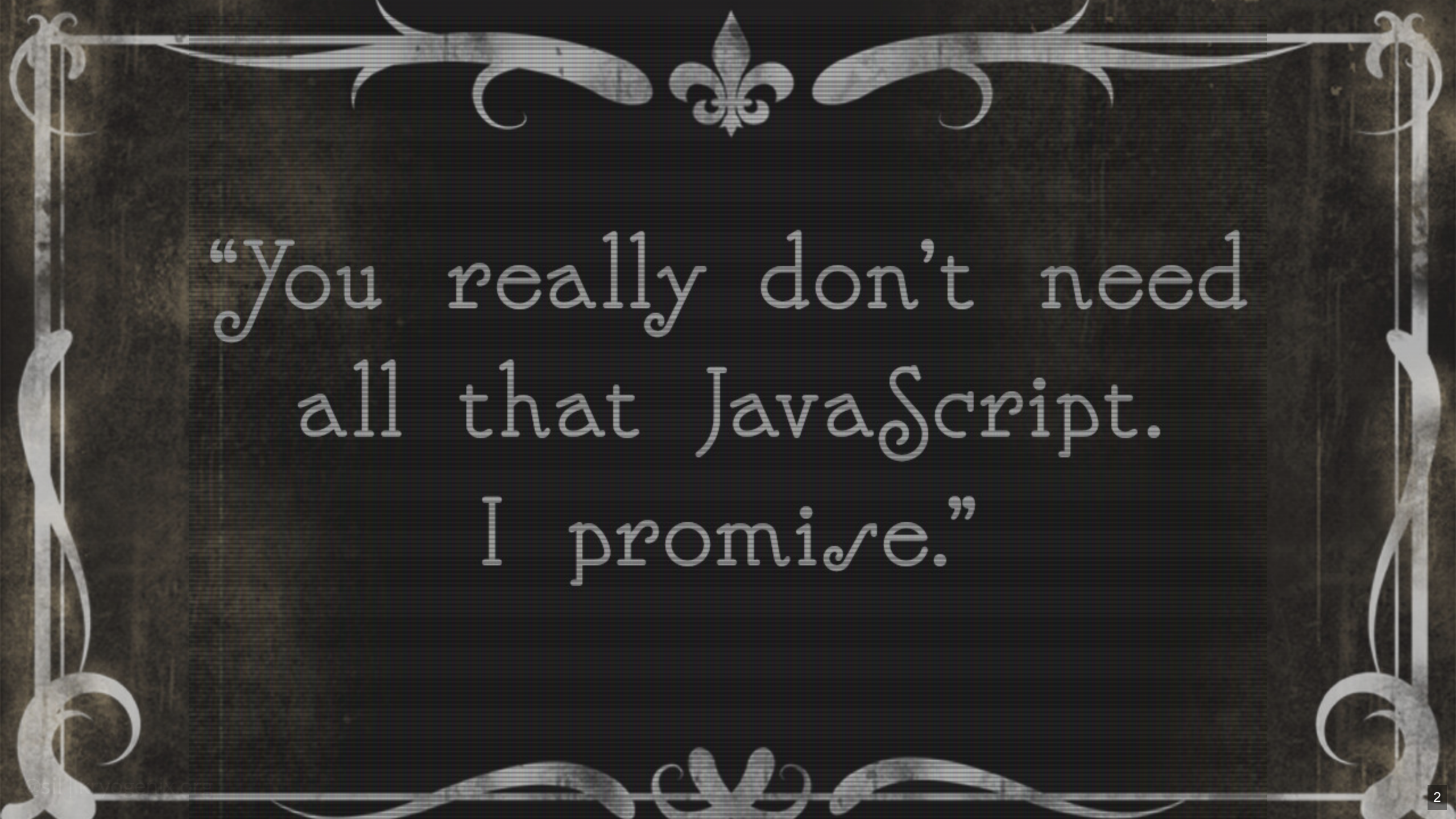


Dear Niamh,

Hello, darling daughter! I thought I'd take this time to write to you while I'm here on my own with a comfortable chair and nobody's listening.

I'm in Romania! It's great here!



“You really don’t need
all that JavaScript.
I promise.”



PERFORMANCE







First meaningful paint

...with *lots* of HTML

Zach Leatherman

@zachleat

Which has a better First Meaningful Paint time?

① a raw 8.5MB HTML file with the full text of every single one of my

27,506 tweets

② a client rendered React site with exactly one tweet on it

(Spoiler: @___lighthouse reports 8.5MB of HTML wins by about 200ms)

6 Sept 2019

twitter.com/zachleat/status/1169998370041208832

Code highlighting

Although the single HTML file is larger with SSR, total transfer size on the SSR version is ~10K smaller.

There is practically zero impact on parsing time even though we know the parsed HTML is twice the size when SSR is in play.

remysharp.com/2019/04/09/code-highlighting-server-or-client

Outsourcing

Alex Russell

@slightlylate

The takeaway here is that you literally can't afford desktop or iphone levels of JS if you're trying to make good web experiences for anyone but the world's richest users

14 Jun 2019

twitter.com/slightlylate/status/1159551977178591233

...SO SERVE LESS...?

THE NETWORK



AVAILABILITY

1.1%

OF PEOPLE AREN'T GETTING JAVASCRIPT ENHANCEMENTS

gds.blog.gov.uk/2013/10/21/how-many-people-are-missing-out-on-javascript-enhancement/





IT'S NOT LIKE THIS

The proportion of people that have explicitly disabled JavaScript or use a browser that doesn't support JavaScript only makes up a small slice of people that don't run JavaScript.

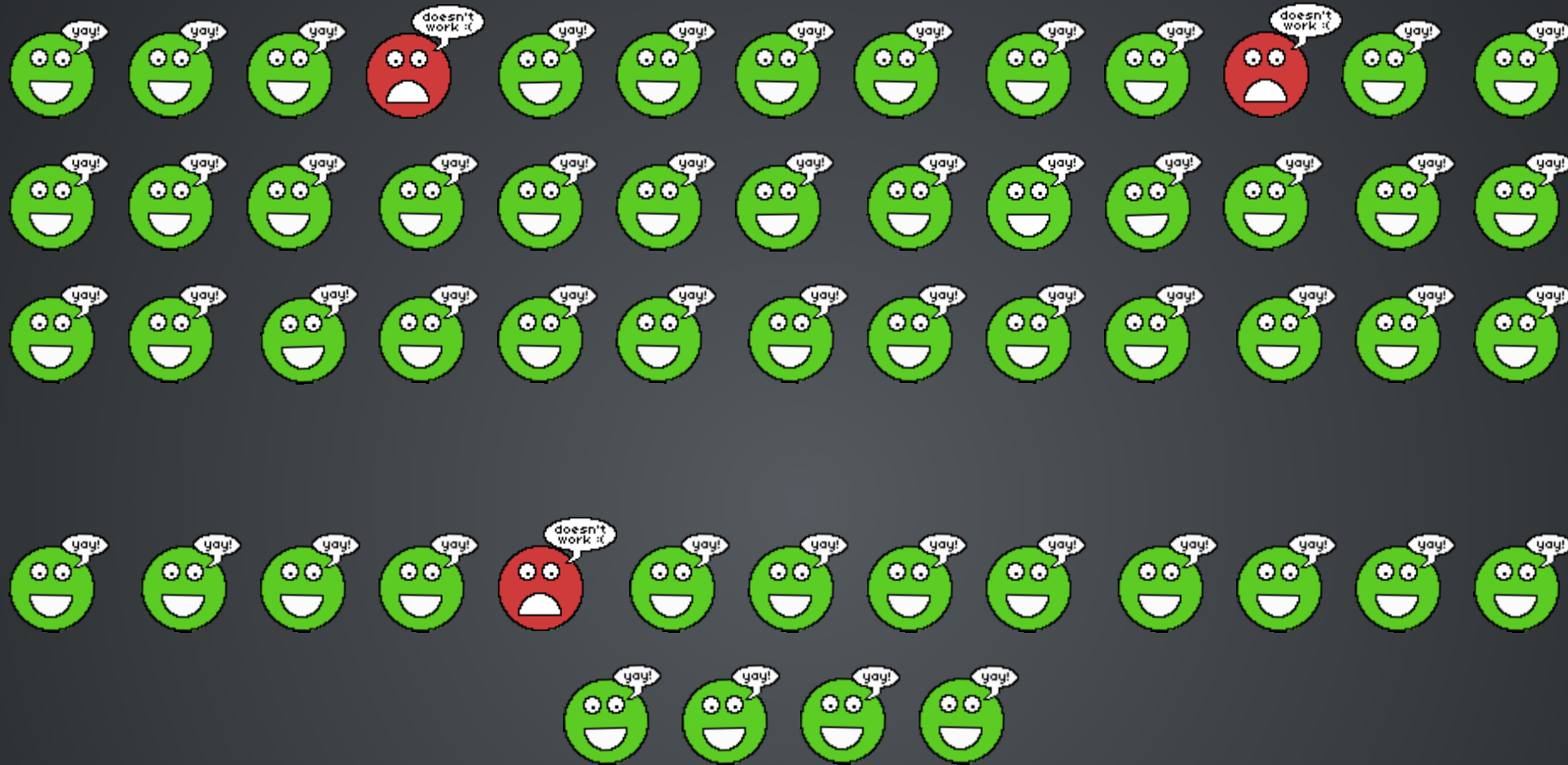
gds.blog.gov.uk/2013/10/21/how-many-people-are-missing-out-on-javascript-enhancement/

Everyone has JavaScript, right?

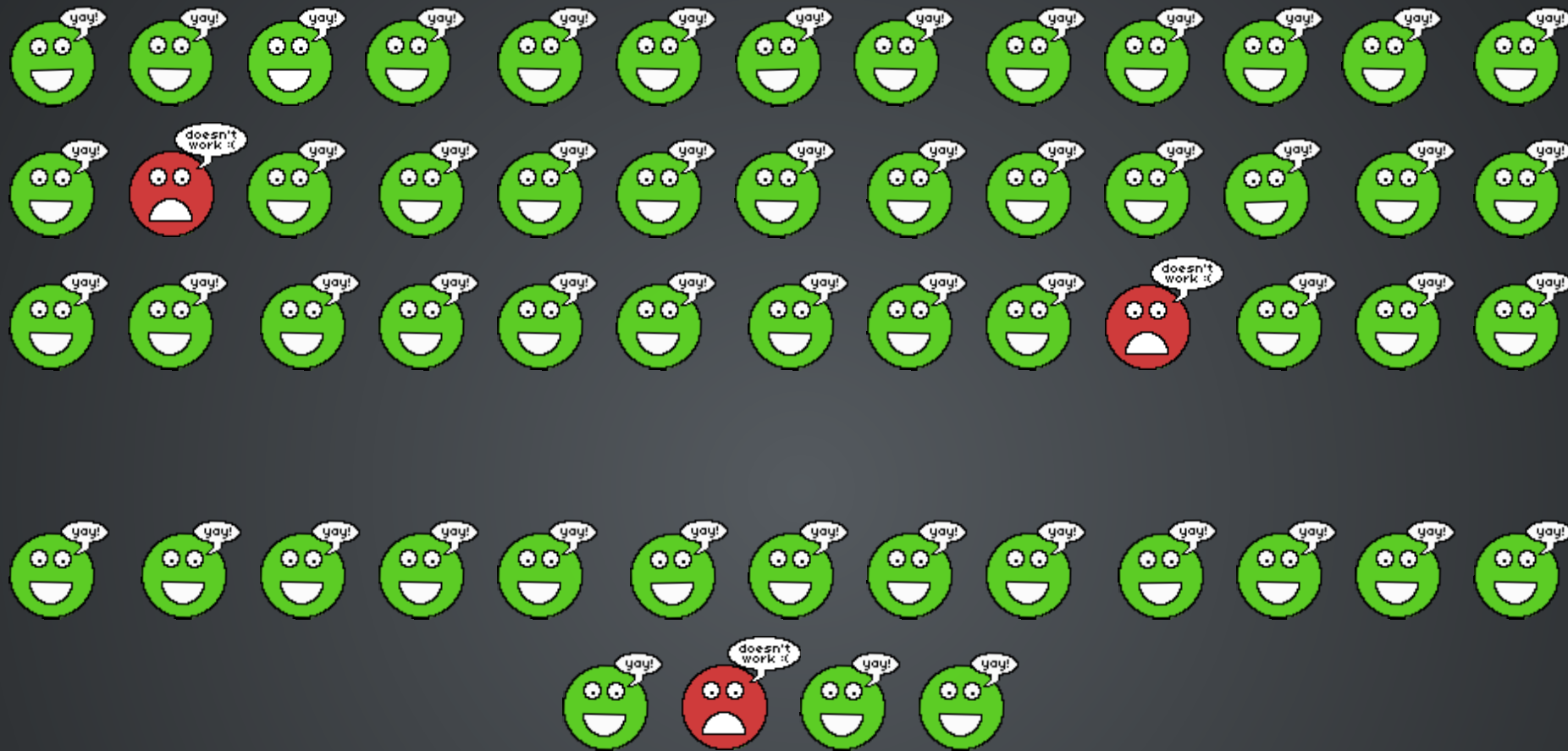
YOUR USER REQUESTS YOUR WEB APP

HAS THE PAGE LOADED YET?

“All your users are non-JS while they're downloading your JS” — [Jake Archibald](#)



IT'S LIKE THIS.



OR WORSE, THIS.

**IT'S UNNECESSARILY
DIFFICULT**

[illegible]

Increasingly, there seems to be a sense of fatigue within our industry. Just when you think you've got a handle on whatever the latest tool or technology is, something new comes out to replace it.

Drew McLellan, 24ways.org/2017/all-that-glitters/

I've discovered how many others have felt similarly, overwhelmed by the choices we have as modern developers, always feeling like there's something we should be doing better.

The web is incredible these days, but I'm not sure every app needs to be a React-based single-page application, because the complexity is so great in a product's infancy.

Owen Williams, [chargd/blog/2019/you-dont-need-that-hipster-web-framework](https://chargd.com/blog/2019/you-dont-need-that-hipster-web-framework)

SASS, JavaScript dialects, NPM, and build tools solve some shortcomings with front-end technologies, but at what cost?

I'd argue that for every hour these new technologies have saved me, they've cost me another in troubleshooting or upgrading the tool due to a web of

*In the meantime, I could have broken out
any text editor and FTP program and
built a full site with plain HTML and CSS
in the time it took me to get it all running.
And the site would have been easier to
maintain in the long-run.*

I can still take a complete beginner and teach them to build a simple webpage with HTML and CSS, in a day. We don't need to talk about tools or frameworks, learn how to make a pull request or drag vast amounts of code onto our computer via npm to make that start. We just need


```

1 <script type="text/javascript">
2 function ajaxLoader(url,id) {
3     if (document.getElementById) {
4         var x = (window.ActiveXObject) ?
5             new ActiveXObject("Microsoft.XMLHTTP") : new XMLHttpRequest();
6     }
7     if (x) { showLoader(); x.onreadystatechange = function() {
8
9         if (x.readyState == 4 && x.status == 200) {
10             el = document.getElementById(id); el.innerHTML = x.responseText; hideLoader();
11         }}
12     x.open("GET", url, true);
13     x.send(null);
14 }
15 </script>
16 <span class="link" onclick="ajaxLoader('page2.html','contentLYR')">page 2</span>

```



```
1 var Component = React.createClass({
2   getInitialState: function() { return {query: ''} },
3   queryChange: function(evt) {
4     this.setState({query: evt.target.value});
5   },
6   handleSearch: function() {
7     window.location = '/search/'+this.state.query+'/some-action';
8   },
9   render: function() {
10    return (
11      <div classname="component-wrapper">
12        <input type="text" value="{this.state.query}">
13        <button onclick="{this.handleSearch()}" classname="button">Search</button>
14      </div>
15    ); } });
```

WHY

- component reuse
- libraries of existing code
- consistent starting point
- organisational structure
- best practices

AFTER THE FACT





HIGH-END MOBILITY

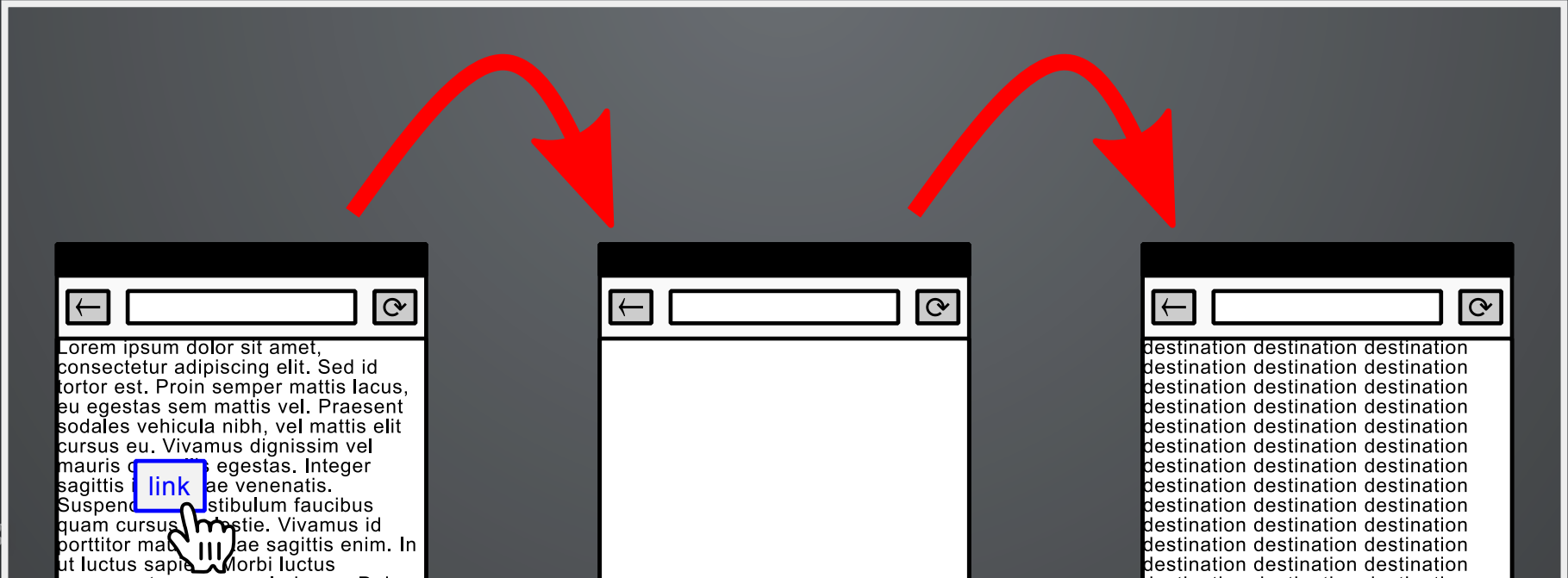


AVERAGE MOBILITY

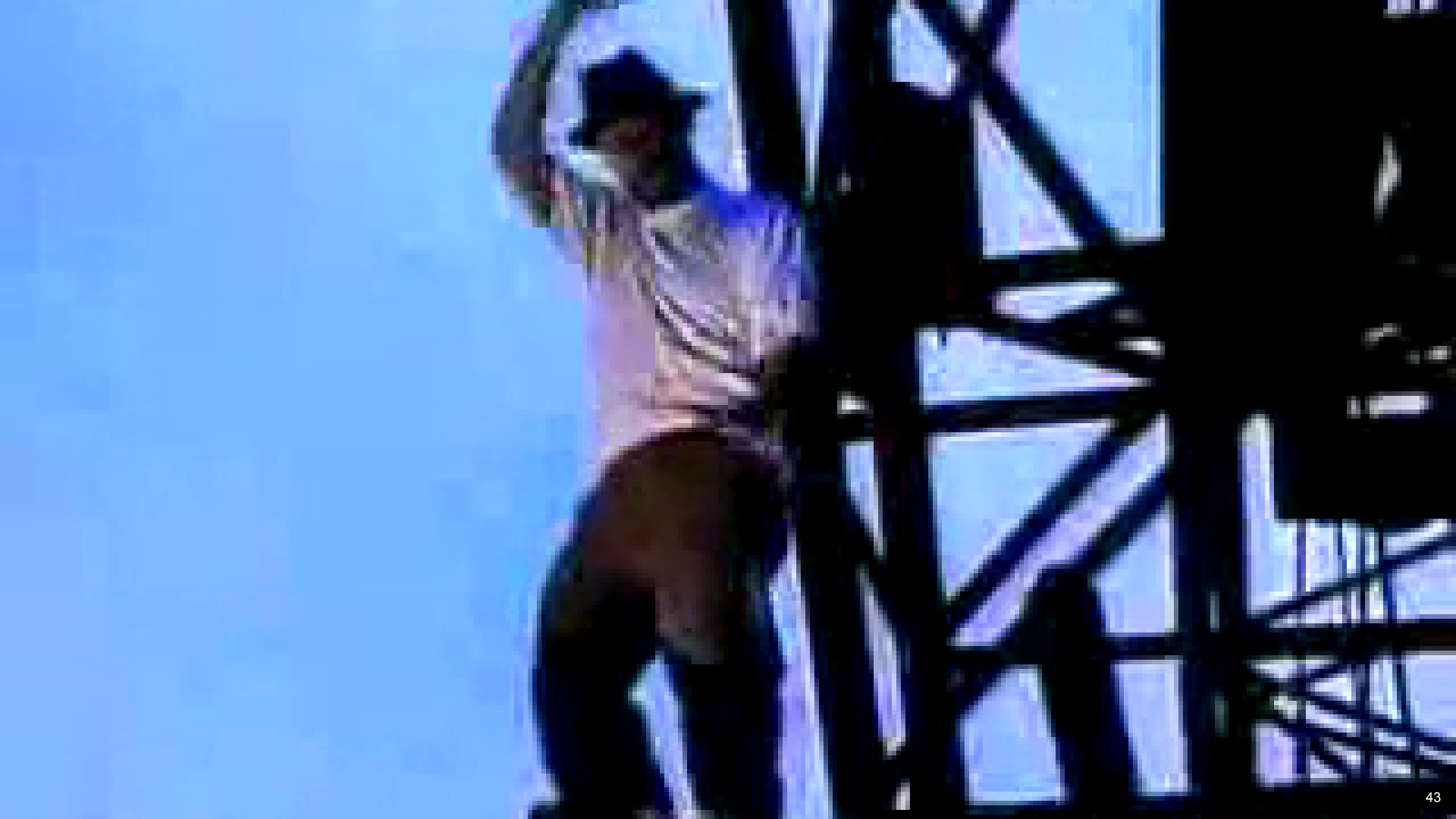


LOW-END MOBILITY





LOSS OF CONTROL



LOSS OF CONTROL

loss of control

`fetch()`

reuse HTML

virtual DOM

different URLs

client-side routing

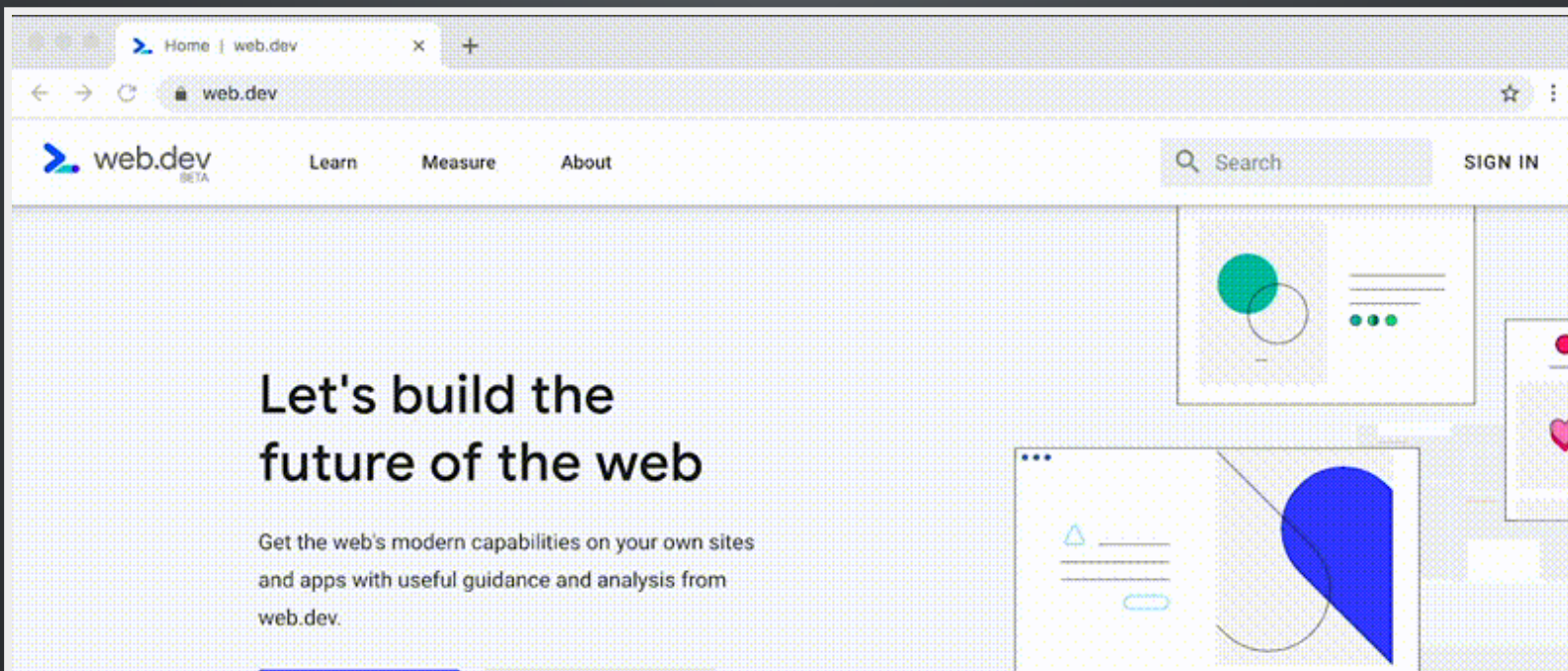
framework





CONTROL LOADING

PORTAL




```
1 const portal = document.createElement('portal');  
2 portal.src = 'https://jscamp.ro';  
  
3 document.body.append(portal);  
4 portal.activate();  
5 ...
```

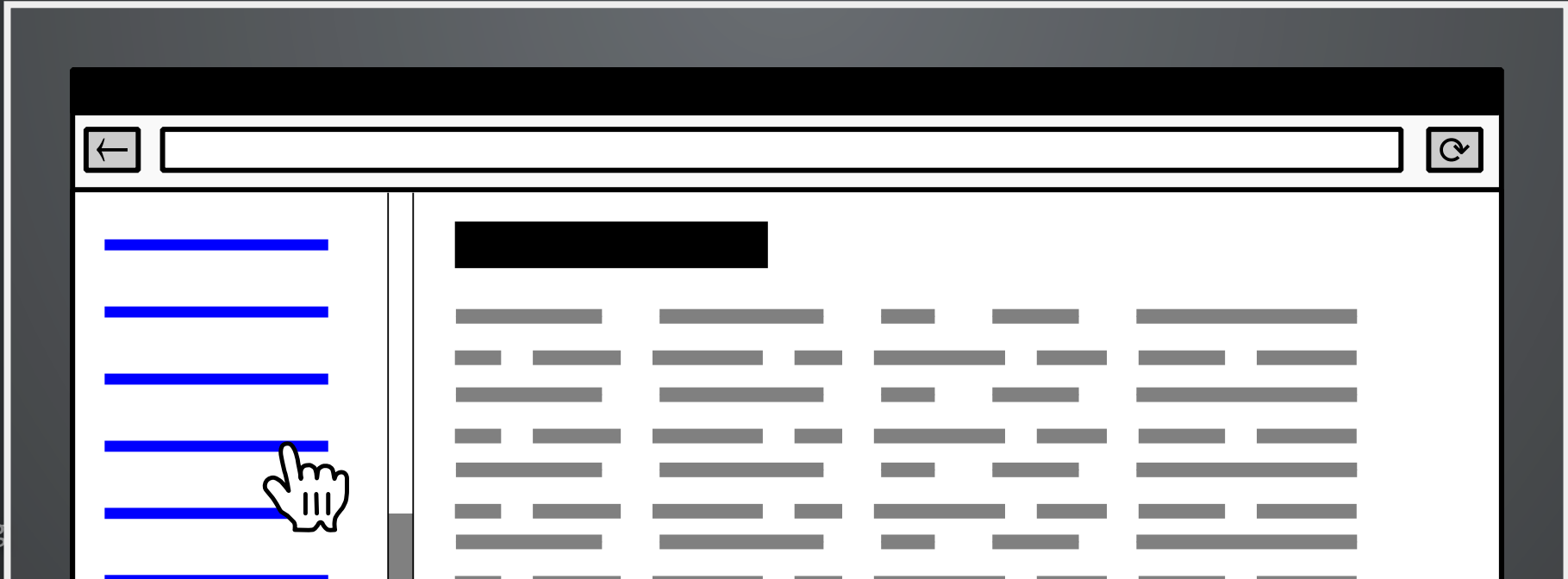
```
1 const portal = document.createElement('portal');  
2 portal.src = 'https://jscamp.ro';  
  
3 document.body.append(portal);  
4 ... do whatever you want ...  
5 portal.activate();
```


so for example, you could **animate it**


```
1 const portal = document.createElement('portal');
2 portal.src = 'https://jscamp.ro';
3 document.body.append(portal);

4 portal.classList.add('expand');
5 portal.addEventListener('transitionend', evt => {
6     portal.activate();
7 });
```

solve actual problems



separate pages?

forgets the scroll position 😞



SPA?

all of this talk up to now 😞

a bunch of hacky JS?



<portal> **and** postMessage

no code in the pages.
no change to the pages.
just enhance how they work.
if your browser supports it.

```

if ("HTMLPortalElement" in window) {
    document.querySelector("aside").addEventListener("click", function(e) {
        if (e.target.nodeName.toLowerCase() == "a") { e.preventDefault(); }
        let sidebarScrollPosition = document.querySelector("aside").scrollTop;
        var portal = document.createElement('portal');
        portal.hidden = true; var u = new URL(e.target.href); u.hash = "";
        portal.addEventListener("message", e => {
            portal.activate({data: {sidebarScrollPosition: sidebarScrollPosition}});
        })
        portal.src = u.href;

        document.body.appendChild(portal);
    }, false);
    window.addEventListener("portalactivate", function(e) {
        if (e.data.sidebarScrollPosition) {
            document.querySelector("aside").scrollTop = e.data.sidebarScrollPosition;
        }
    }, false)
    if (window.portalHost) {
        window.portalHost.postMessage({done: true, frm: location.pathname});
    }
}

```





Add React to a Website

Use as little or as much React as you need.

React has been designed from the start for gradual adoption, and **you can use as little or as much React as you need**. Perhaps you only want to add some “sprinkles of interactivity” to an existing page. React components are a great way to do that.

The majority of websites aren’t, and don’t need to be, single-page apps. With **a few lines of code and no build tooling**, try React in a small part of your website. You can then either gradually expand its presence, or keep it contained to a few dynamic widgets.

Systems status

Web server	
Back end	
Debruceifier	

Using [coderdiaz/vue-status-indicator](https://github.com/coderdiaz/vue-status-indicator).

HTML: smarter than it used to be

<datalist>

```
<input list="thelist">  
<datalist id="thelist">  
  <option>Bruce</option>  
  
  <option>Stuart</option>  
</datalist>
```

Best dressed speaker?



<fieldset disabled>

disable/enable

Enter your name

Oranges ☐ Apples ☐

CSS scroll-snap

```
parent { scroll-snap-type: y mandatory; }  
child { scroll-snap-align: start; }
```



Ana Tudor

@anatudor

Just because you don't understand a simple little CSS solution, it doesn't mean it's "weird"/ "crazy". Goes the other way as well. Adding a ton of extra elements just for the sake of having a pure CSS solution when updating a value from the JS would suffice is just silly.

14 Sept 2019

twitter.com/anatudor/status/1172891778032316416

stay in touch

flaki

@slsoftworks

Not knowing stuff is normal.

Not using new and shiny tech is okay.

Feeling bad for all of these is common.

You are not alone.

But you are also completely and perfectly fine, and

you

are

adequate.



15 Sept 2019

twitter.com/slsoftworks/status/1173224870362021888

fight the ~~good~~ right fight

the great thing about standards

THE **LEADING** EDGE

The word "LEADING" is rendered in a bold, red, 3D sans-serif font. A thick, glossy red liquid, resembling blood, is dripping down from the bottom of the letters. The drips are of varying lengths and thicknesses, creating a sense of movement and gravity. The words "THE" and "EDGE" are in a white, 3D sans-serif font, positioned on either side of "LEADING". The entire graphic is set against a dark gray background.

So, O darling daughter, you asked, and
you probably got a longer answer than
you bargained for!

Still, now you know!

It's valuable, what we do as an
industry.

So, O darling daughter, you asked, and
you probably got a longer answer than
you bargained for!

Still, now you know!

It's valuable, what we do as an
industry.

We have the power to bring knowledge to
the whole world.

Still, now you know!

It's valuable, what we do as an industry.

We have the power to bring knowledge to the whole world.

To connect people together (when they want it),

It's valuable, what we do as an industry.

We have the power to bring knowledge to the whole world.

To connect people together (when they want it),

to be the greatest repository of information that's ever been known.

We have the power to bring knowledge to the whole world.

To connect people together (when they want it),

to be the greatest repository of information that's ever been known.

I want to keep it that way.

To connect people together (when they want it),

to be the greatest repository of information that's ever been known.

I want to keep it that way.

That's what I think is important about the web.

