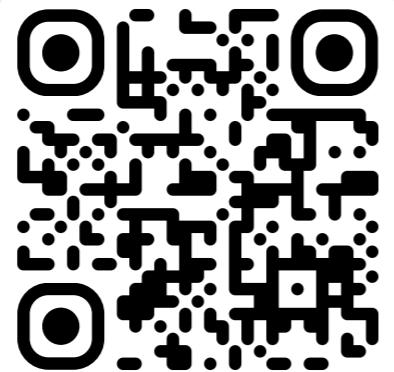


Hi, I'm Elly Loel (she/her)



elly.to/wdc24-slides



<e//y>

Hope you've all had a good break and you're feeling caffeinated and energised.

It's not long now until happy hour. So hang in there!

This is my first time at Web Directions.

It's been an awesome first day so far!

I'd love to say a massive thank you and round of applause to the Web Directions team.

This is actually my first time at an in-person tech conference, so it's also my first time talking at one.

Hi I'm Elly, my pronouns are she/her.

I'm a tall white trans woman with blonde hair.

If you want to follow along with the slides they are available up on Notist at elly.to/wdc24-slides that's E L L Y dot T O slash W D C 2 4 dash S L I D E S.

Okay! Let's get into it.

Front-end developments identity crisis

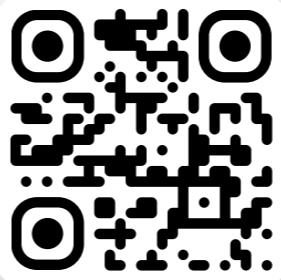
<e//y>

So this afternoon I'm going to be talking to you about Front-end developments identity crisis.

What would you call someone who:

- Builds **accessible** and **usable** UIs
- Uses **HTML**, **CSS**, and  JS
- Collaborates with **designers**, **developers**, and **testers**

 elly.to/wdc1



<e//y>

We're gonna start off with a couple of questions.

If you can all scan this QR code or head to the link elly.to/wdc1.

So the question is:

"What would you call someone who builds accessible and usable UIs; uses HTML, CSS, and  JS; and collaborates with designers, developers, and testers?"

Hopefully we'll start to see some answers coming through

...

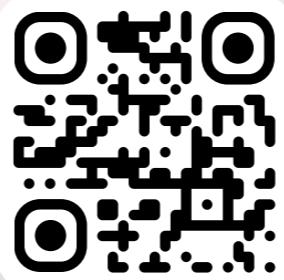
Those were some really interesting answers!

Same question, but *doesn't*:

- Elf 🧝 Develop and optimise **application logic**
- Elf 🧝 Integrate **data** sources and services
- Elf 🧝 Test **performance** and **functionality**
- Elf 🧝 Manage **JS tooling** and **infrastructure**



elly.to/wdc1



<e//y>

Now, I've got a follow up question for you all.

Same drill, scan the QR code or head to the link elly.to/wdc1.

So the second question is:

"What would you call someone who does all of the things from the previous question but **doesn't**: develop and optimise application logic; integrate data sources and services; ensure performance and functionality with tests; and manage JS tooling and infrastructure?"

So there seems to be some confusion about what exactly this role is.

So, let's work through it by looking at some of my experiences.

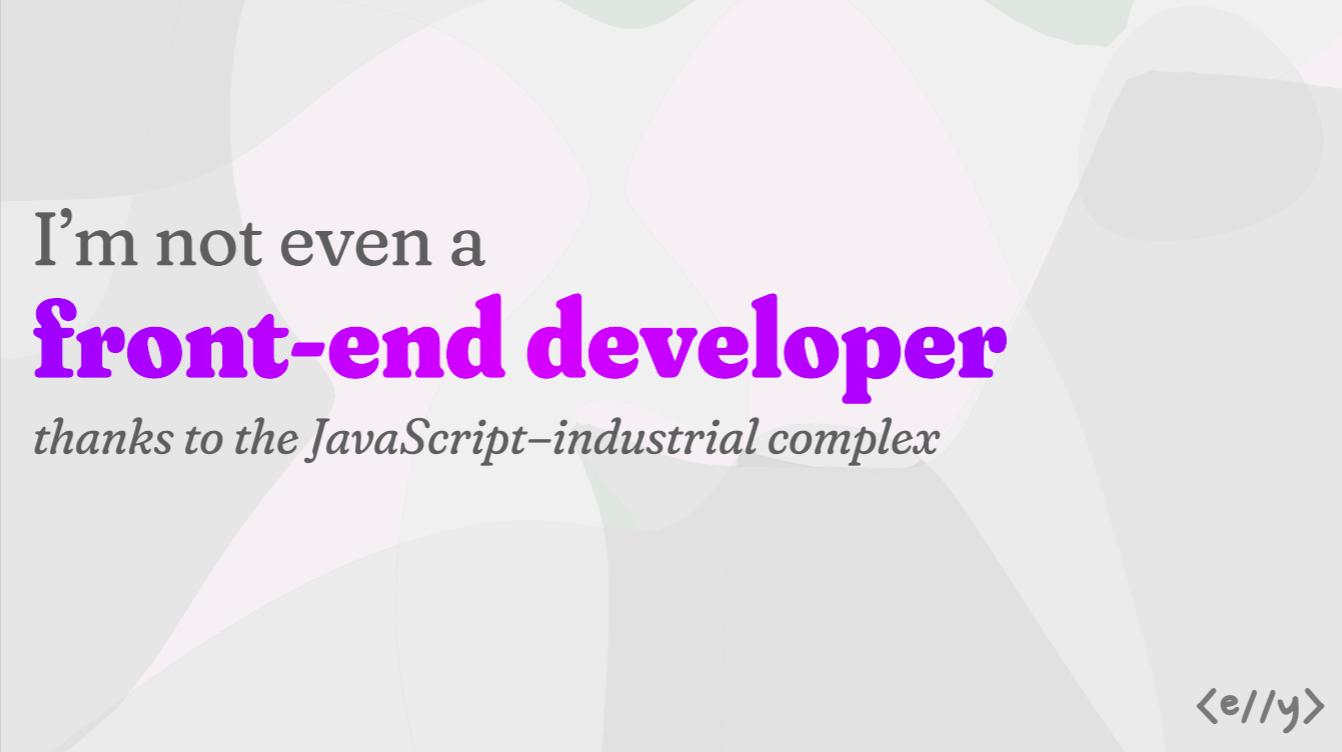


I'm not a
full-stack developer

regardless of what my last job title says

<e//y>

I'm not a full-stack developer, regardless of what my last job title says.



I'm not even a
front-end developer

thanks to the JavaScript-industrial complex

<e//y>

I'm not even a front-end developer, thanks to the JavaScript-industrial complex.



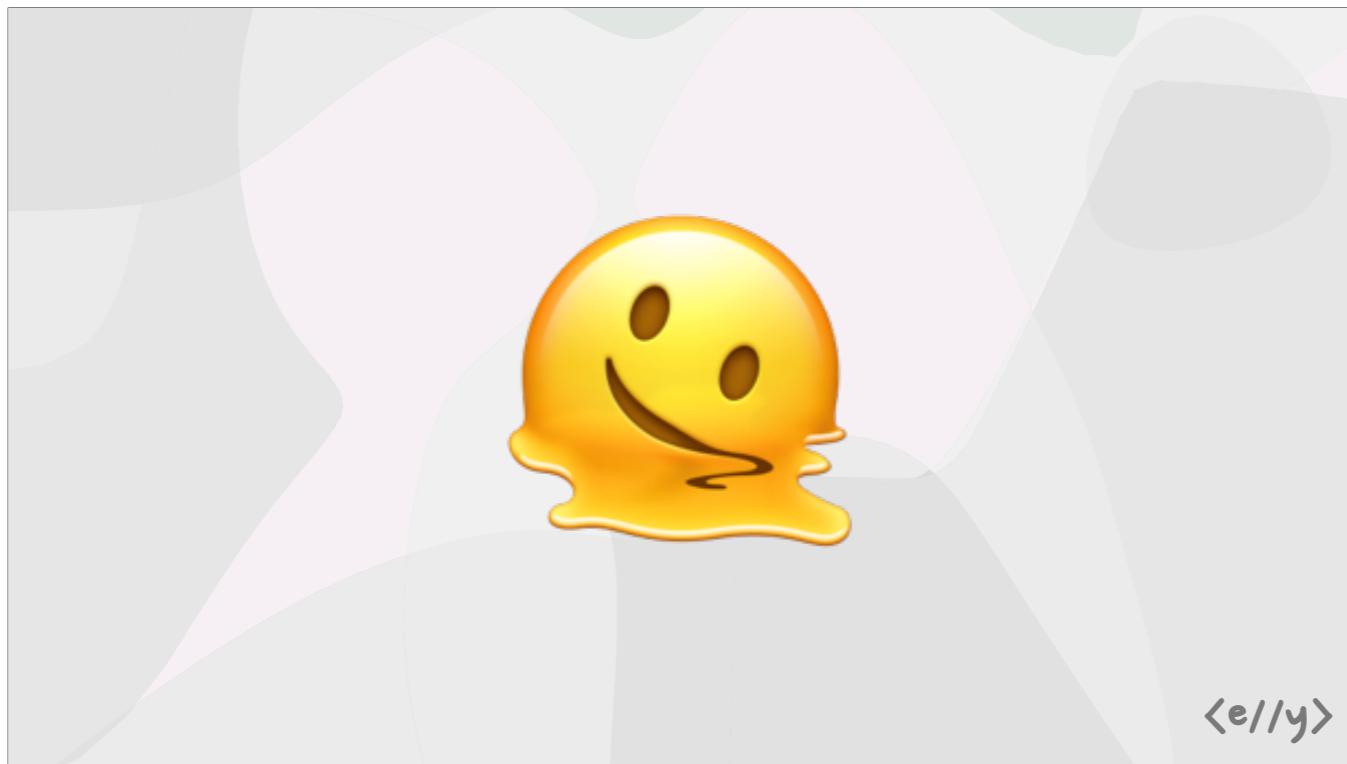
I'm a front-of-the-front-end developer, but that's too long.



So, I'm a
web designer

<e//y>

So, I'm a web designer.



sigh 😔 wooh!



That was densely packed,
so let's break it down

<e//y>

That was a lot.
Very densely packed.
Let's break it down.

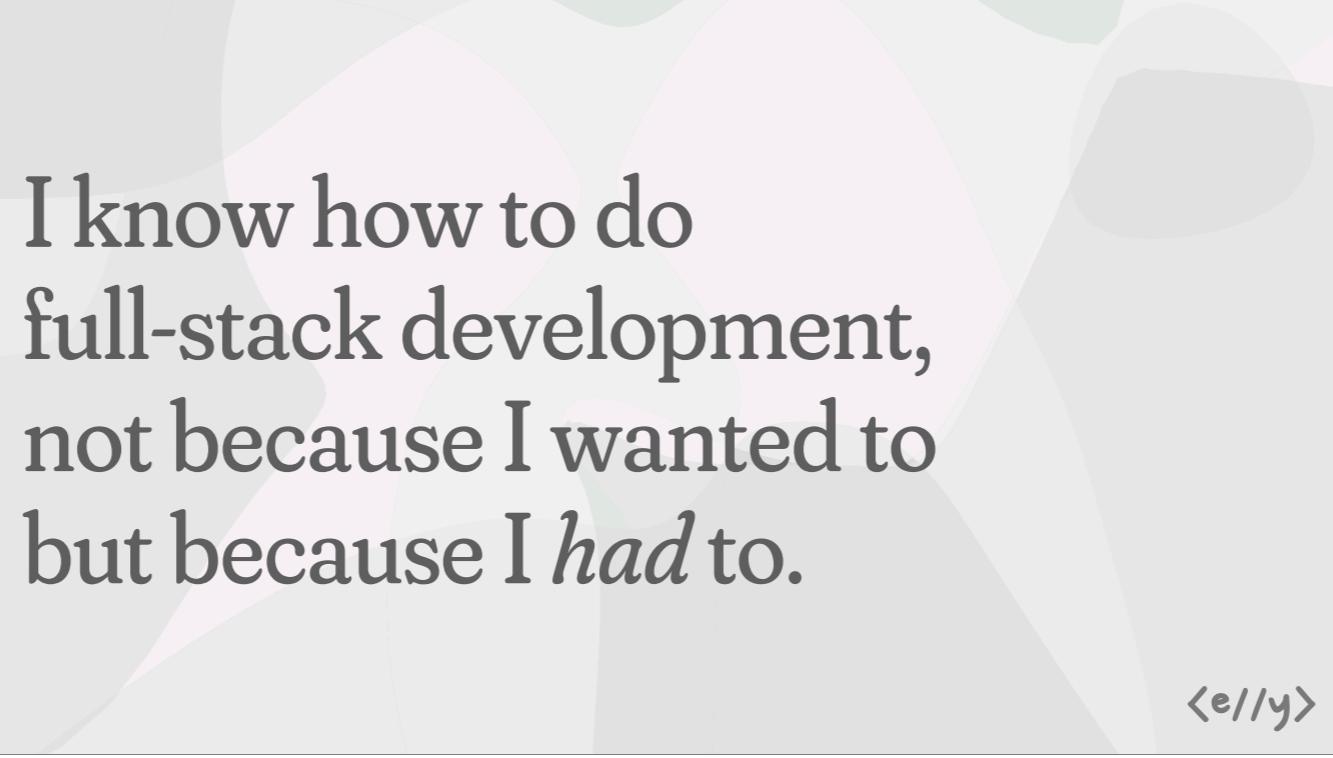


I'm not a
full-stack developer

regardless of what my last job title says

<e//y>

I'm not a full-stack developer, regardless of what my last job title says.



I know how to do
full-stack development,
not because I wanted to
but because I *had* to.

<e//y>

I know how to do full-stack development.
Not because I wanted to but because I had to.



“developer”

`<e//y>`

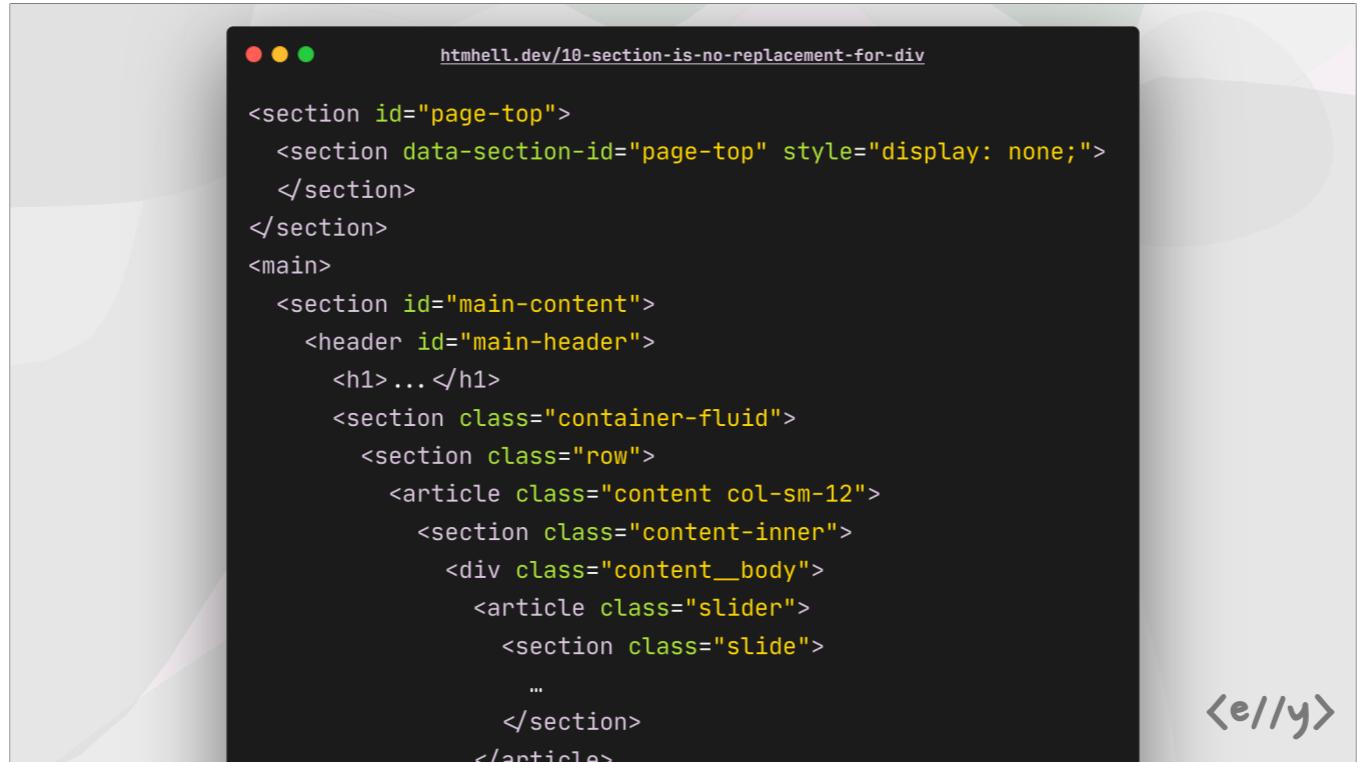
My title was “developer”.
And so was everyone else’s.



~~-front-end~~

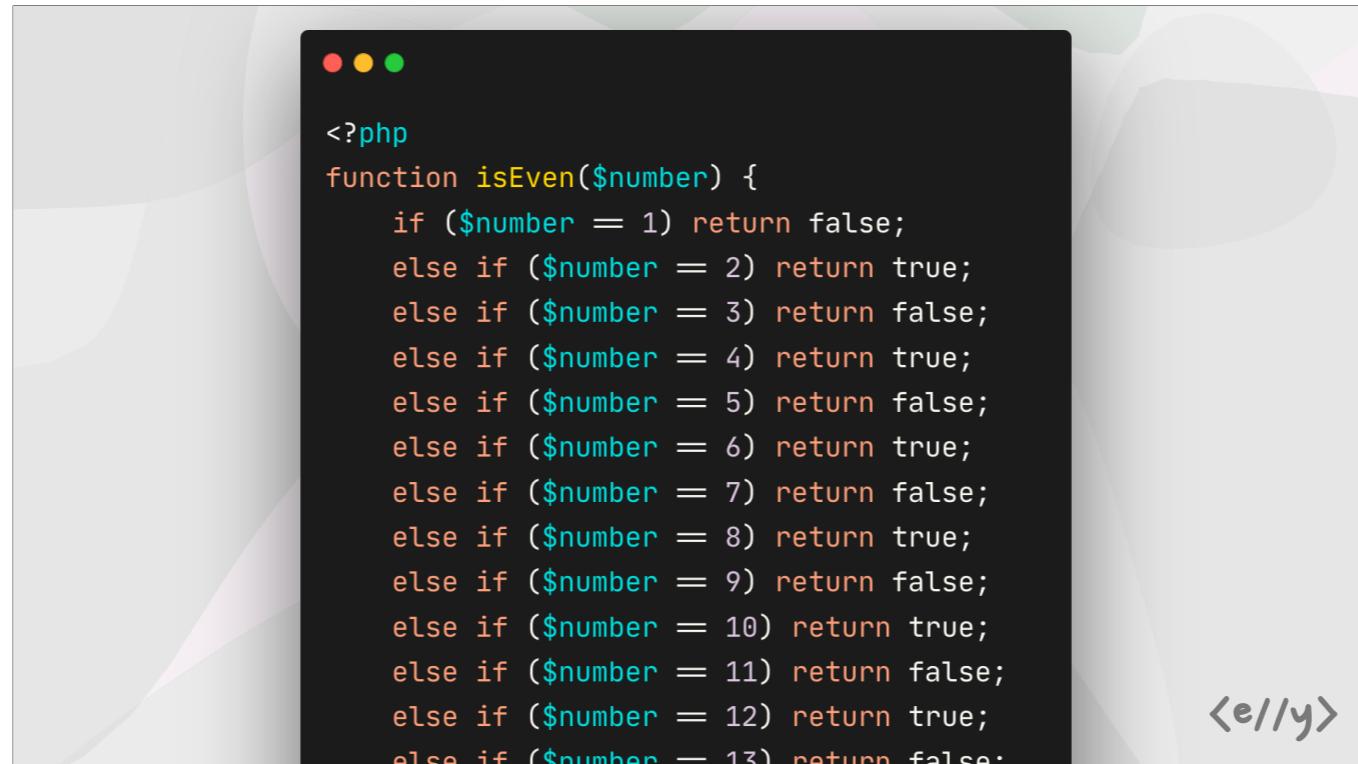
~~-back end~~

According to our titles there were no front-end or back-end developers.
We had to do it all.

A screenshot of a web browser window titled "htmhell.dev/10-section-is-no-replacement-for-div". The page displays a highly nested and complex HTML structure. It starts with a main `<section id="page-top">`, which contains a section with `data-section-id="page-top"` and `style="display: none;"`. This is followed by a `<main>` section, then a `<section id="main-content">`, and a `<header id="main-header">`. Inside the header is an `<h1>.... Below the header is a <section class="container-fluid">, which contains a <section class="row">. Inside the row is an <article class="content col-sm-12">, which contains a <section class="content-inner">, a <div class="content__body">, and an <article class="slider">. The slider section contains a <section class="slide"> with three dots (...). The entire structure is enclosed in a large <article> tag. In the bottom right corner of the browser window, there is a small icon or watermark that says "<e//y>".

```
<section id="page-top">
  <section data-section-id="page-top" style="display: none;">
    </section>
</section>
<main>
  <section id="main-content">
    <header id="main-header">
      <h1>...</h1>
      <section class="container-fluid">
        <section class="row">
          <article class="content col-sm-12">
            <section class="content-inner">
              <div class="content__body">
                <article class="slider">
                  <section class="slide">
                    ...
                  </section>
                </article>
              </div>
            </section>
          </article>
        </section>
      </header>
    </section>
  </main>
</section>
````

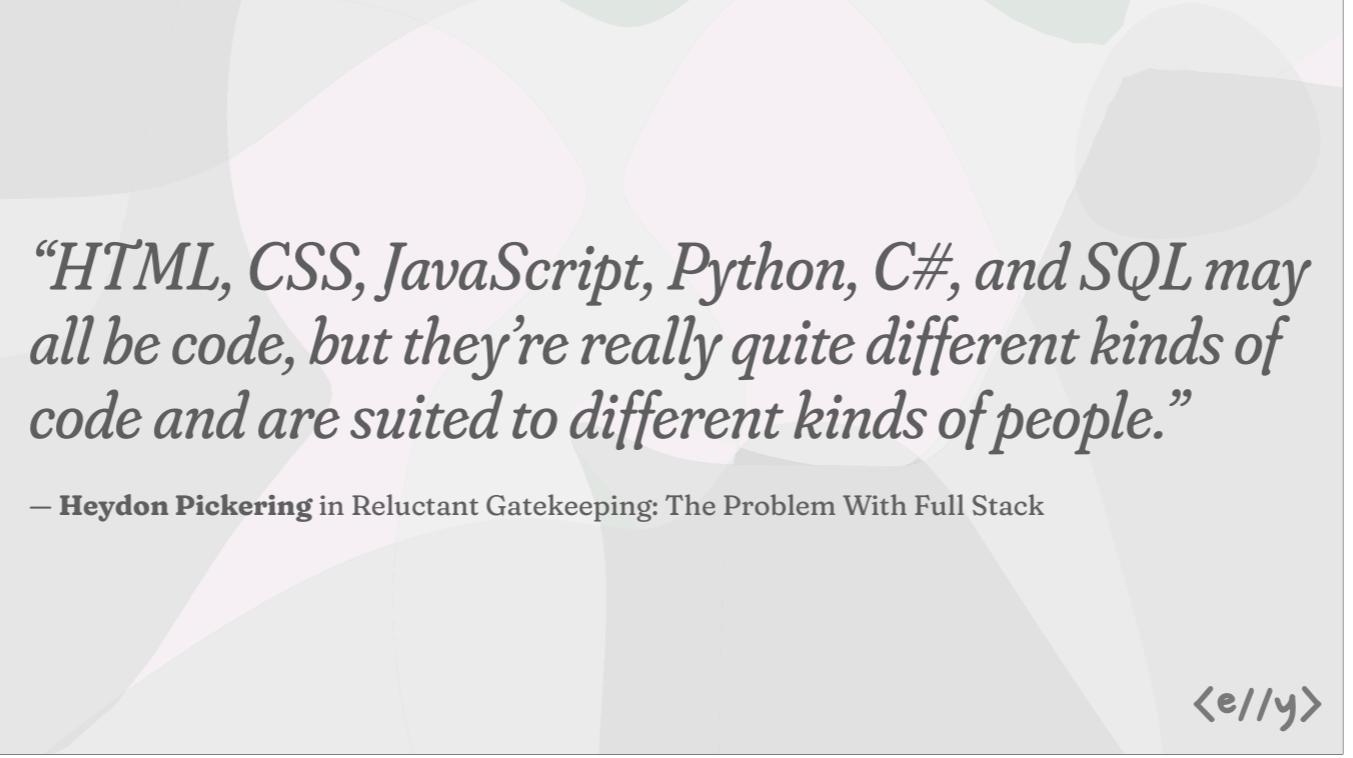
This led to lots of bad HTML and CSS from back-end developers.



```
<?php
function isEven($number) {
    if ($number == 1) return false;
    else if ($number == 2) return true;
    else if ($number == 3) return false;
    else if ($number == 4) return true;
    else if ($number == 5) return false;
    else if ($number == 6) return true;
    else if ($number == 7) return false;
    else if ($number == 8) return true;
    else if ($number == 9) return false;
    else if ($number == 10) return true;
    else if ($number == 11) return false;
    else if ($number == 12) return true;
    else if ($number == 13) return false.
```

← e//y →

And lots of bad Node and PHP from front-end developers.



“HTML, CSS, JavaScript, Python, C#, and SQL may all be code, but they’re really quite different kinds of code and are suited to different kinds of people.”

— Heydon Pickering in Reluctant Gatekeeping: The Problem With Full Stack

<ell>

In Reluctant Gatekeeping: The Problem With Full Stack Heydon Pickering says:

“HTML, CSS, JavaScript, Python, C#, and SQL may all be code, but they’re really quite different kinds of code and are suited to different kinds of people.”



“The value you want from a CSS expert is their CSS, not their JavaScript, so it’s absurd to make JavaScript a requirement.”

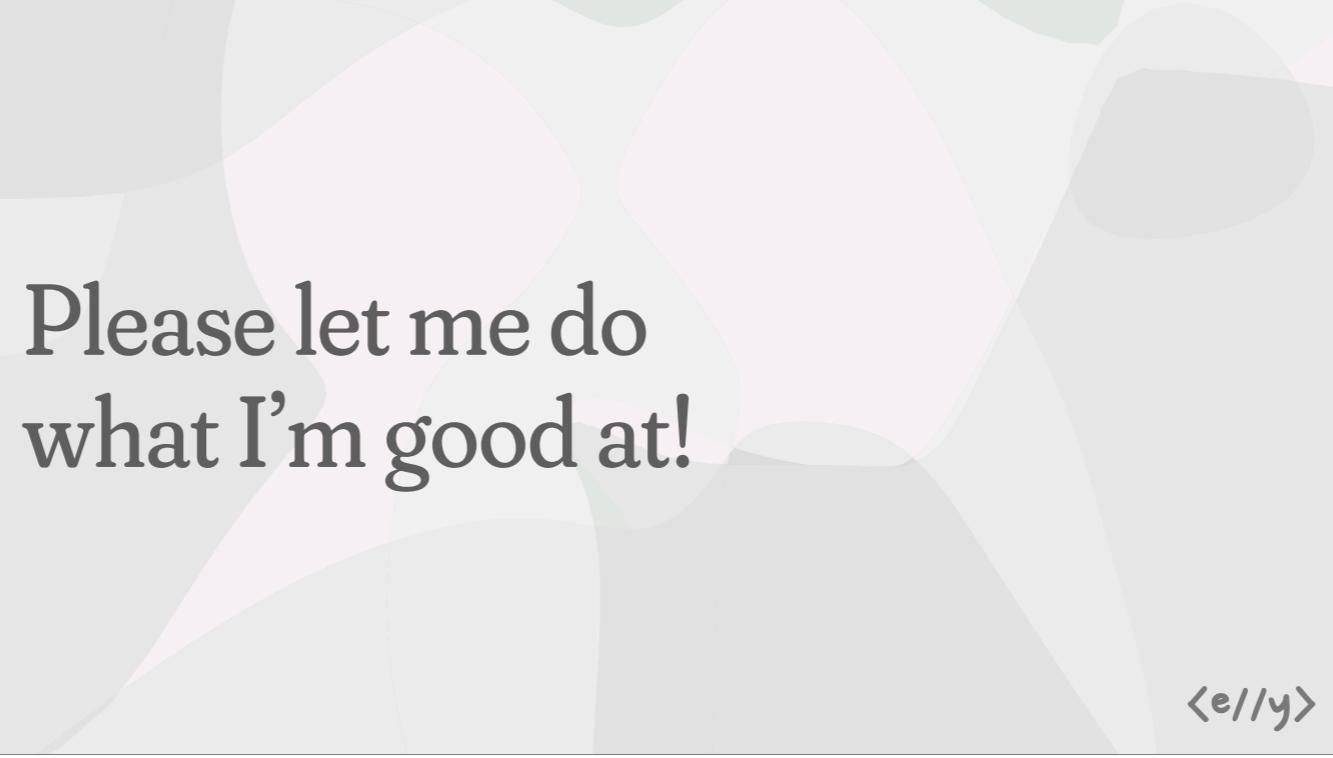
— Heydon Pickering in Reluctant Gatekeeping: The Problem With Full Stack

`<ell>`

“The value you want from a CSS expert is their CSS, not their JavaScript, so it’s absurd to make JavaScript a requirement.”



Just because I know how to use
Docker, Node.js, PHP, SQL, AWS, Linux, Serverless, CI/CD, etc.
doesn't mean I should have to.



Please let me do
what I'm good at!

<e//y>

Please let me do what I'm good at!

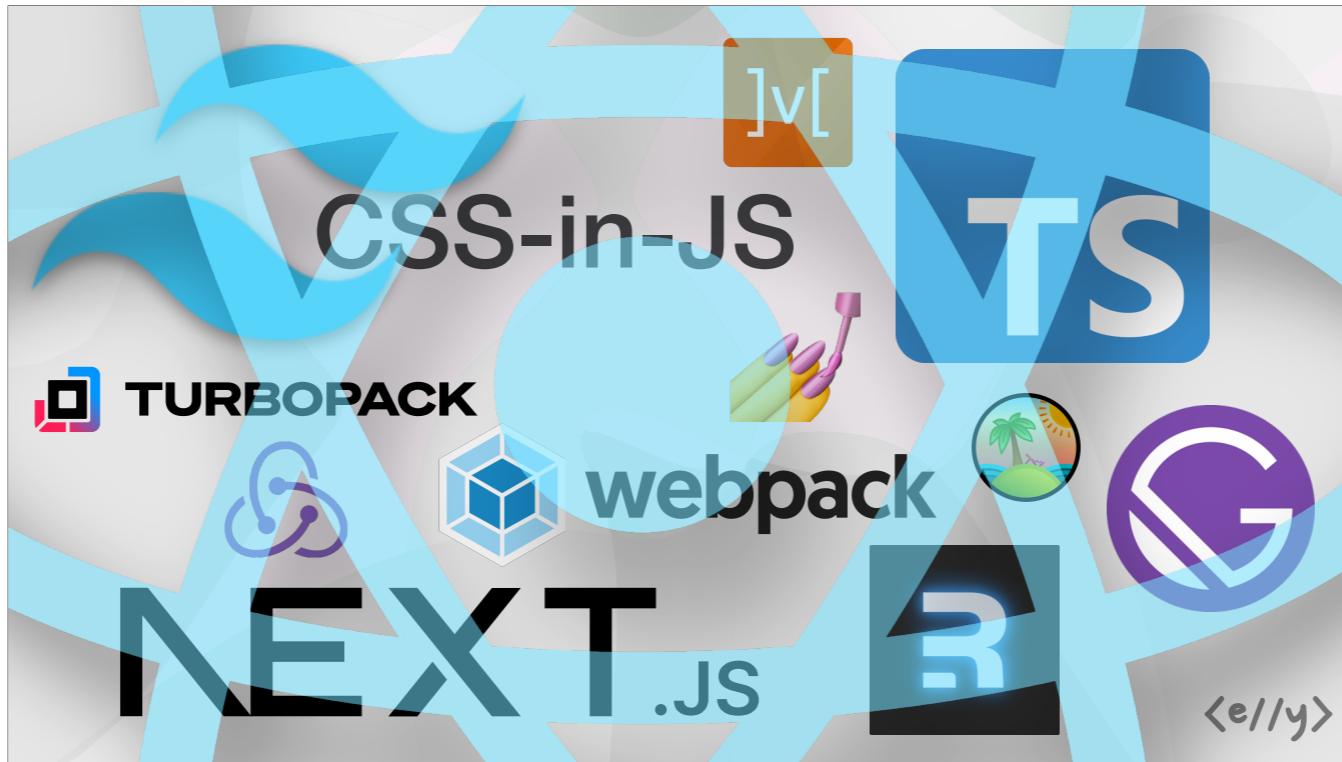


I'm not even a
front-end developer

thanks to the JavaScript-industrial complex

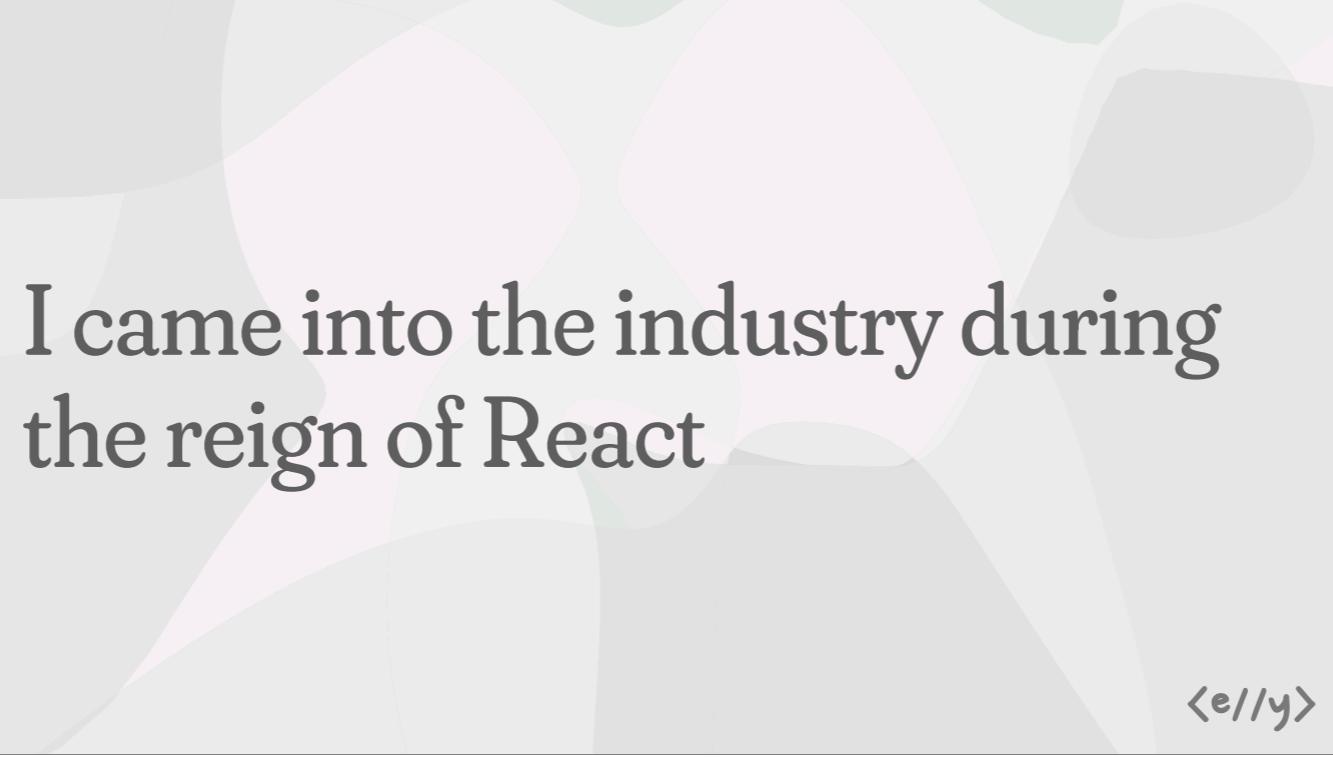
<e//y>

I'm not even a front-end developer, thanks to the JavaScript-industrial complex.



In this section whenever I refer to React please feel free to replace it with any of the following:

- Next.js (what people mean when they say React)
- Tailwind (what people mean when they say CSS)
- Webpack
- CSS-in-JS
- TypeScript
- etc.



I came into the industry during
the reign of React

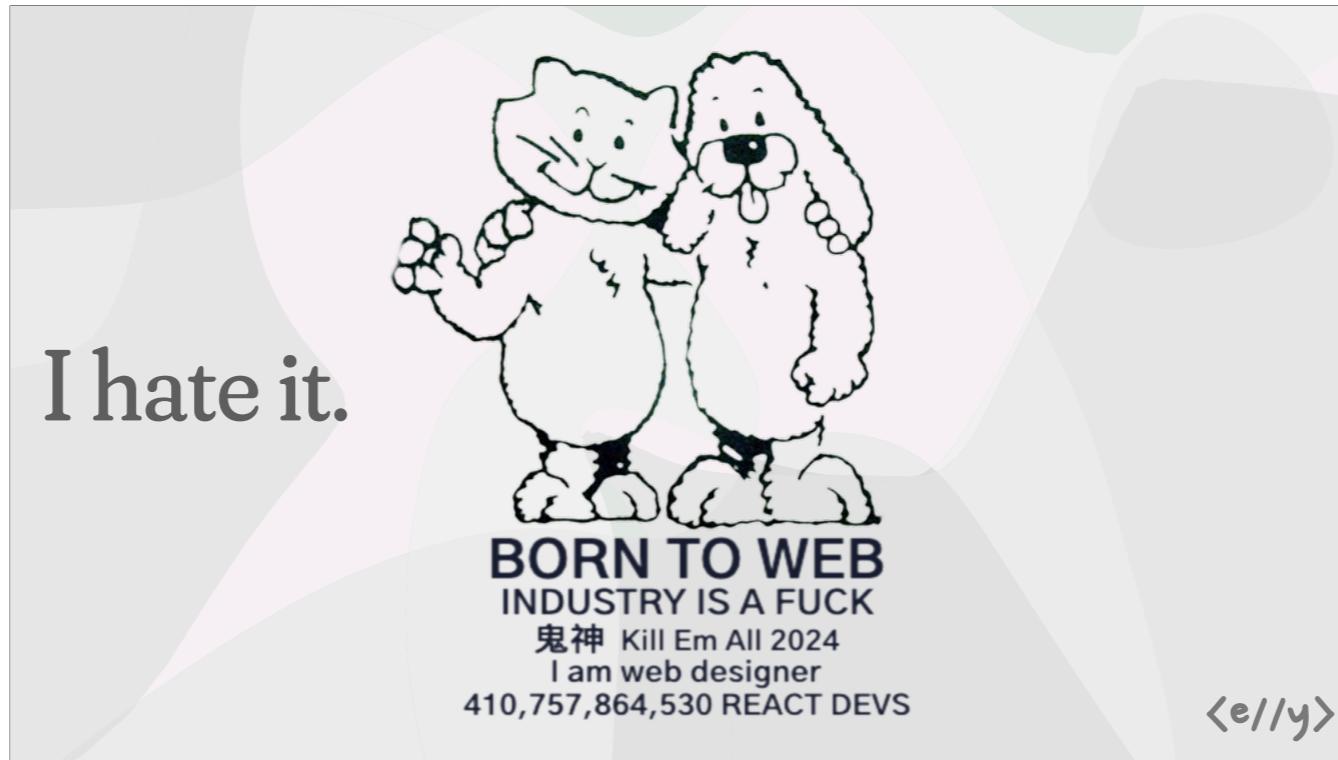
<e//y>

I came into the industry during the reign of React.
I know React well.
It was all I knew for a while.



`<e//y>`

I rushed past HTML and CSS on my way to learning React.
I didn't understand they were the main building blocks of the web.



I hate it.

There are so many things wrong with React.

The cult surrounding it, the company making it, its countless footguns, its bloat, its incompatibility with web platform features—the list goes on.



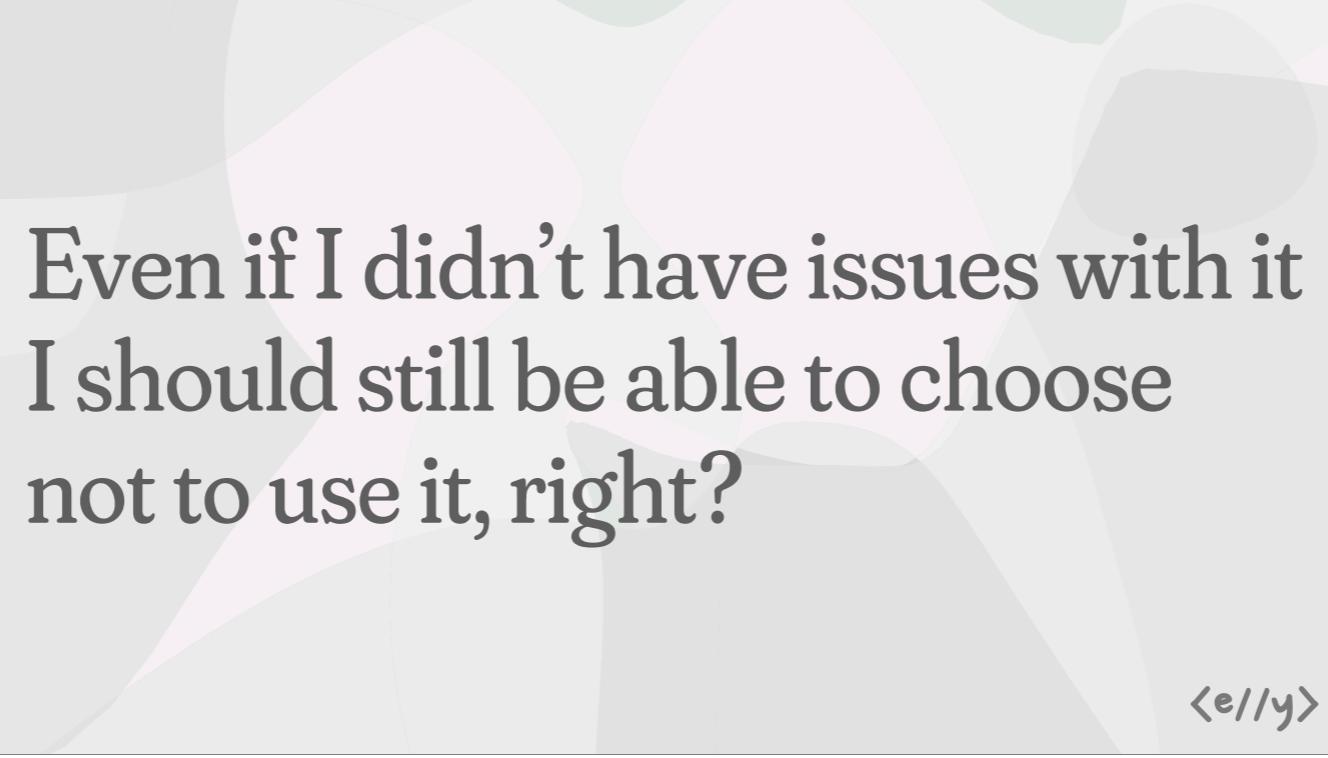
“Not only are new services being built to a self-defeatingly low UX and performance standard, existing experiences are pervasively re-developed on unspeakably slow, JS-taxed stacks.”

— **Alex Russell** in The Market for Lemons

`<e//y>`

In The Market for Lemons Alex Russell says:

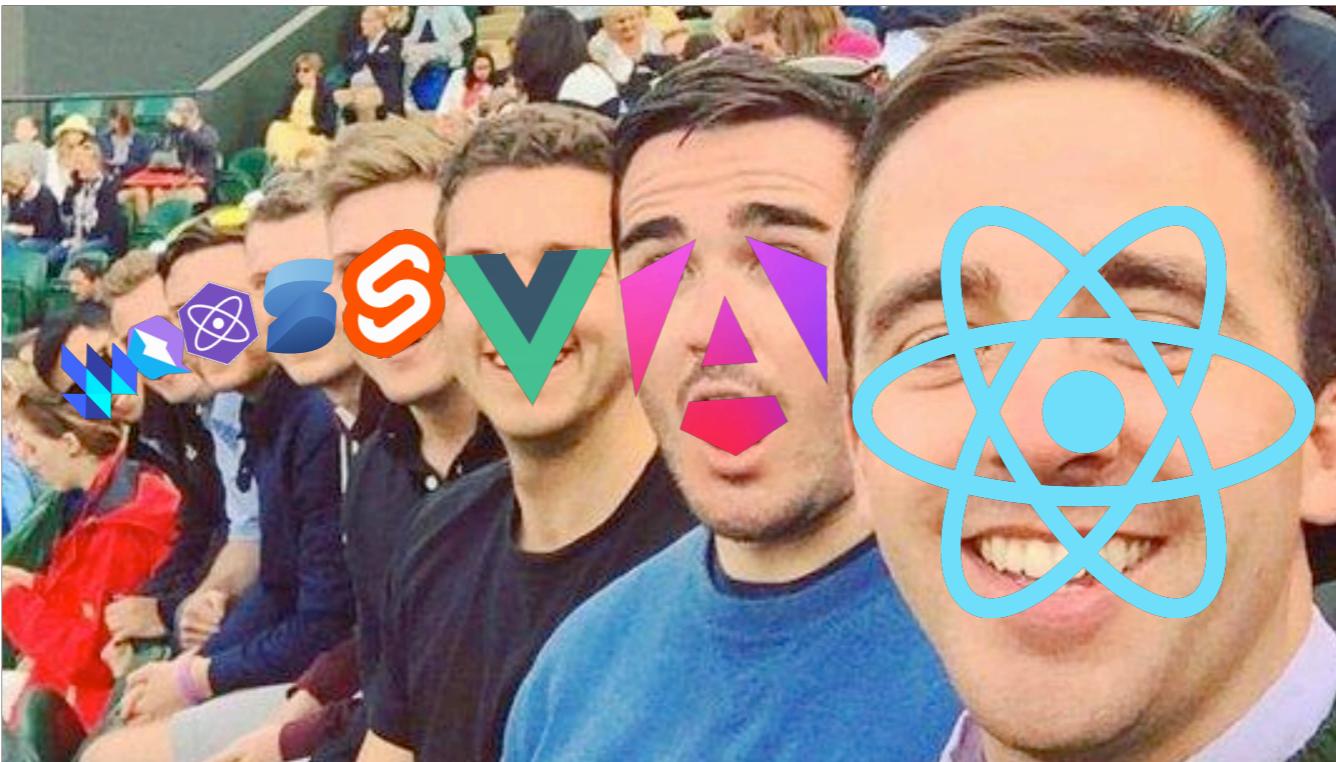
“Not only are new services being built to a self-defeatingly low UX and performance standard, existing experiences are pervasively re-developed on unspeakably slow, JS-taxed stacks.”



Even if I didn't have issues with it
I should still be able to choose
not to use it, right?

<e//y>

Even if I didn't have issues with it I should still be able to choose not to use it, right?

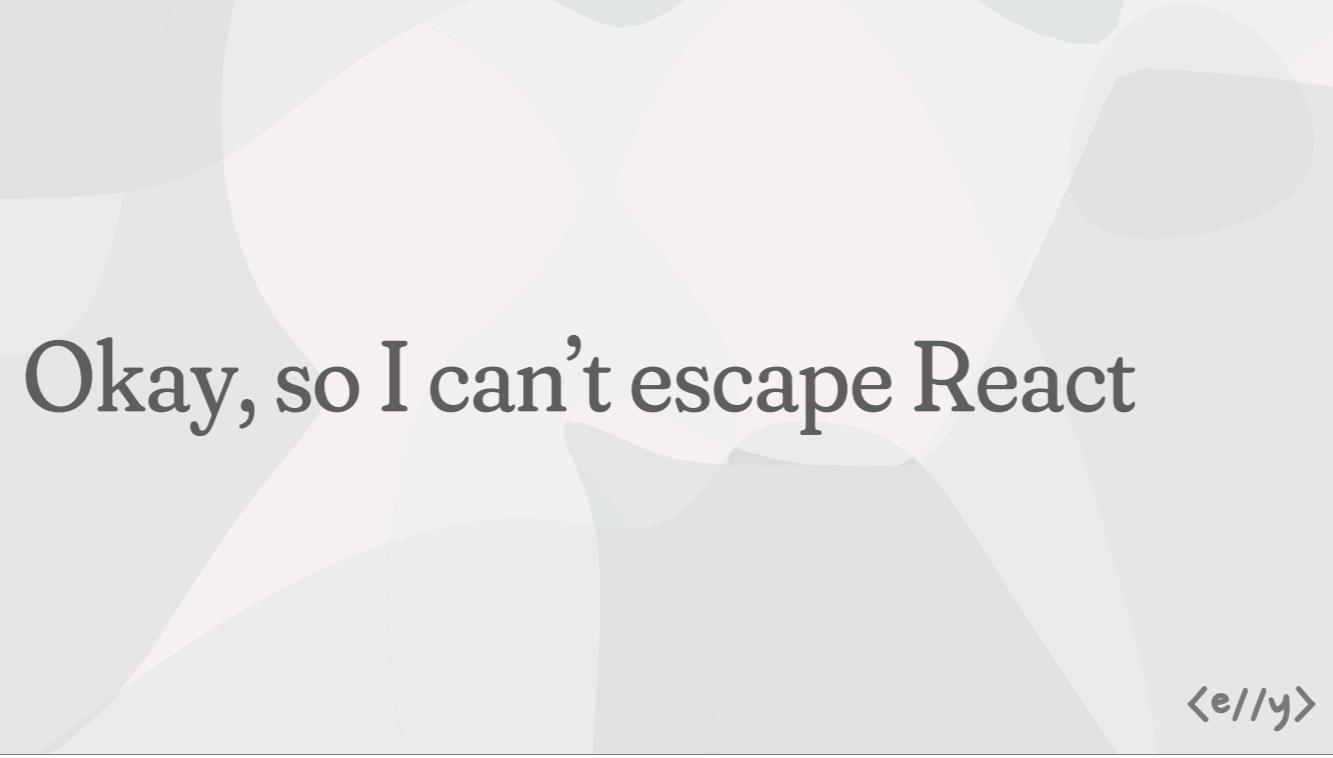


Wrong, sorry!

There is no choice anymore, I can't escape it.

React is so pervasive that almost every job is using it.

On the rare occasion that they're not using it, they're using something like it.



Okay, so I can't escape React

<ell>

Okay, so I can't escape React but surely I could just work on something that doesn't touch React.
Like for example, CSS!



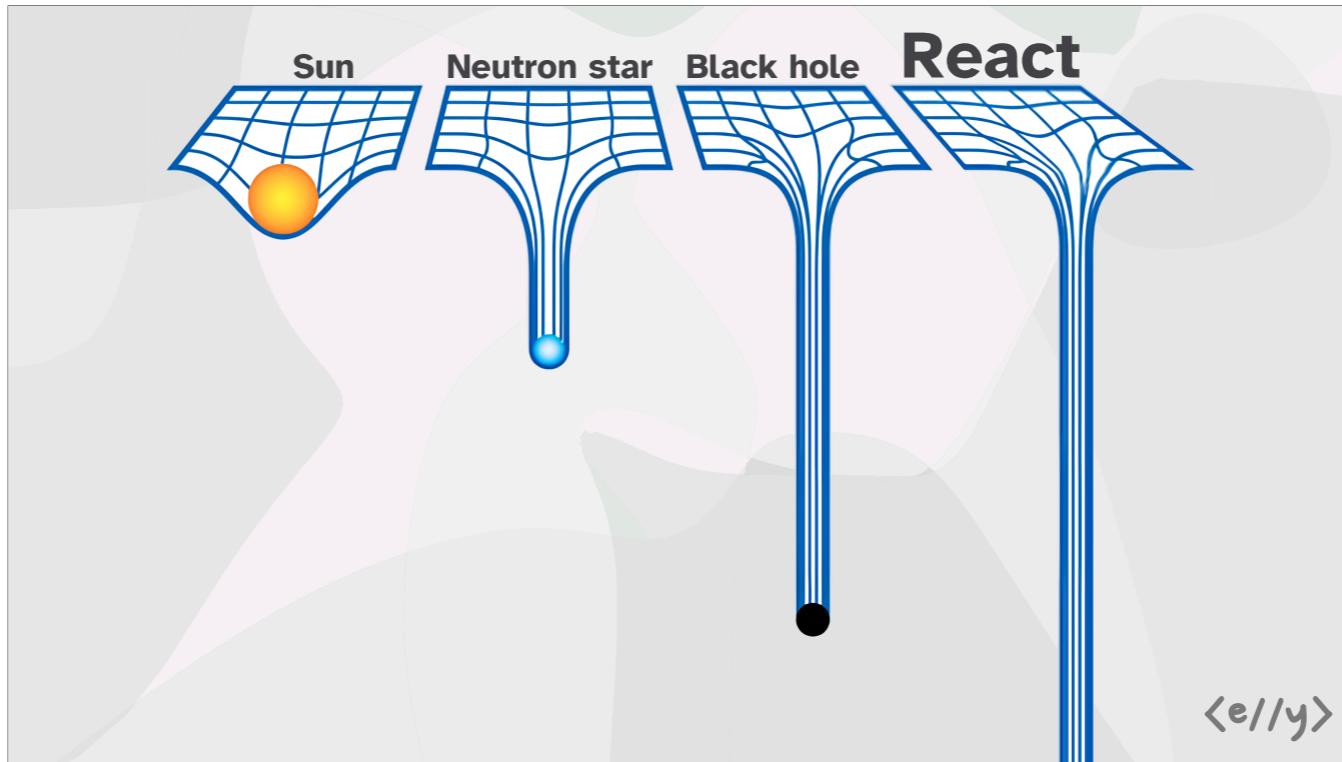
Oh wait, no.

That's all done with CSS modules, CSS-in-JS, Tailwind, or whatever the next new way they come up with to write CSS without actually writing CSS.

If you can't beat 'em,
join 'em, I guess? 🤷‍♀️

<e//y>

If you can't beat 'em, join 'em, I guess? 🤷‍♀️



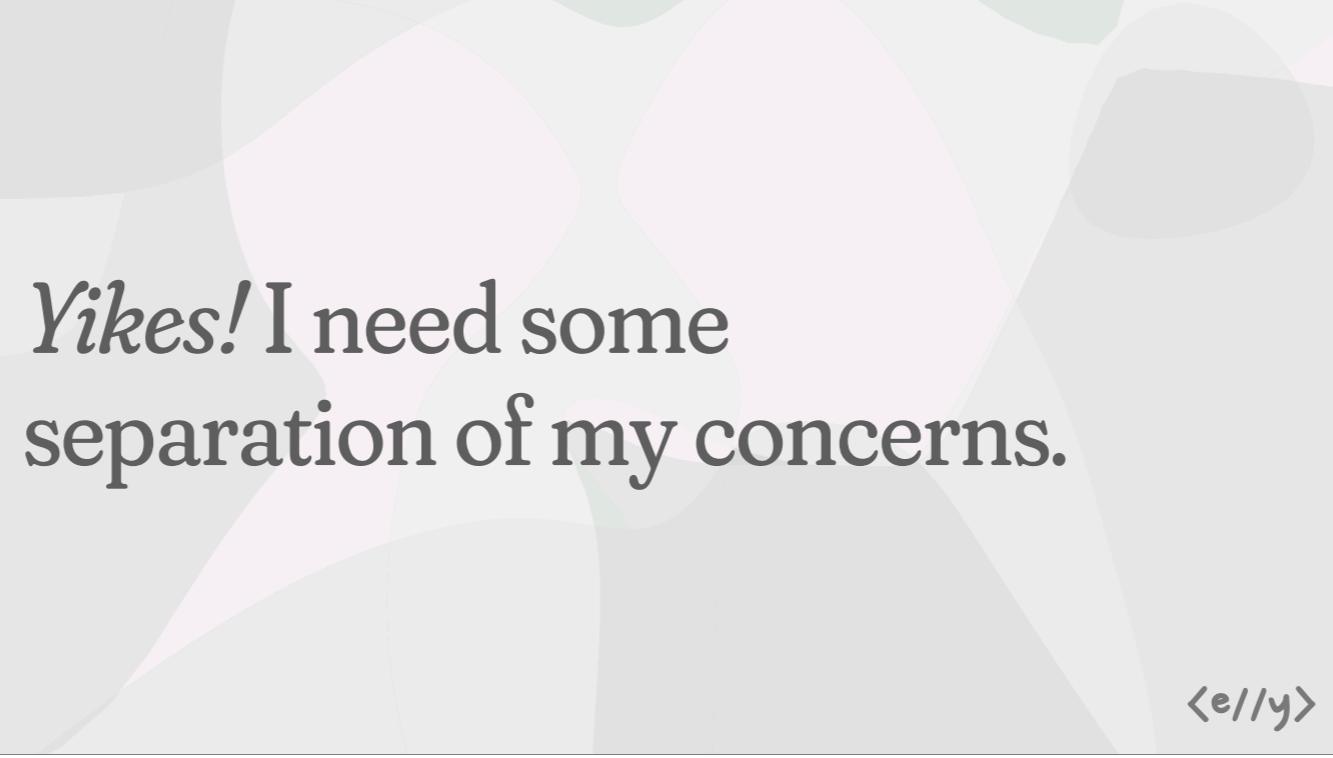
The issue is, React—like the supermassive black hole it is—pulls everything towards it.

Towards more JavaScript.

I start out only touching the HTML, CSS, and a bit of JavaScript for DOM manipulation.

But before I know it I'm in there making API calls, transforming data, and writing middleware.

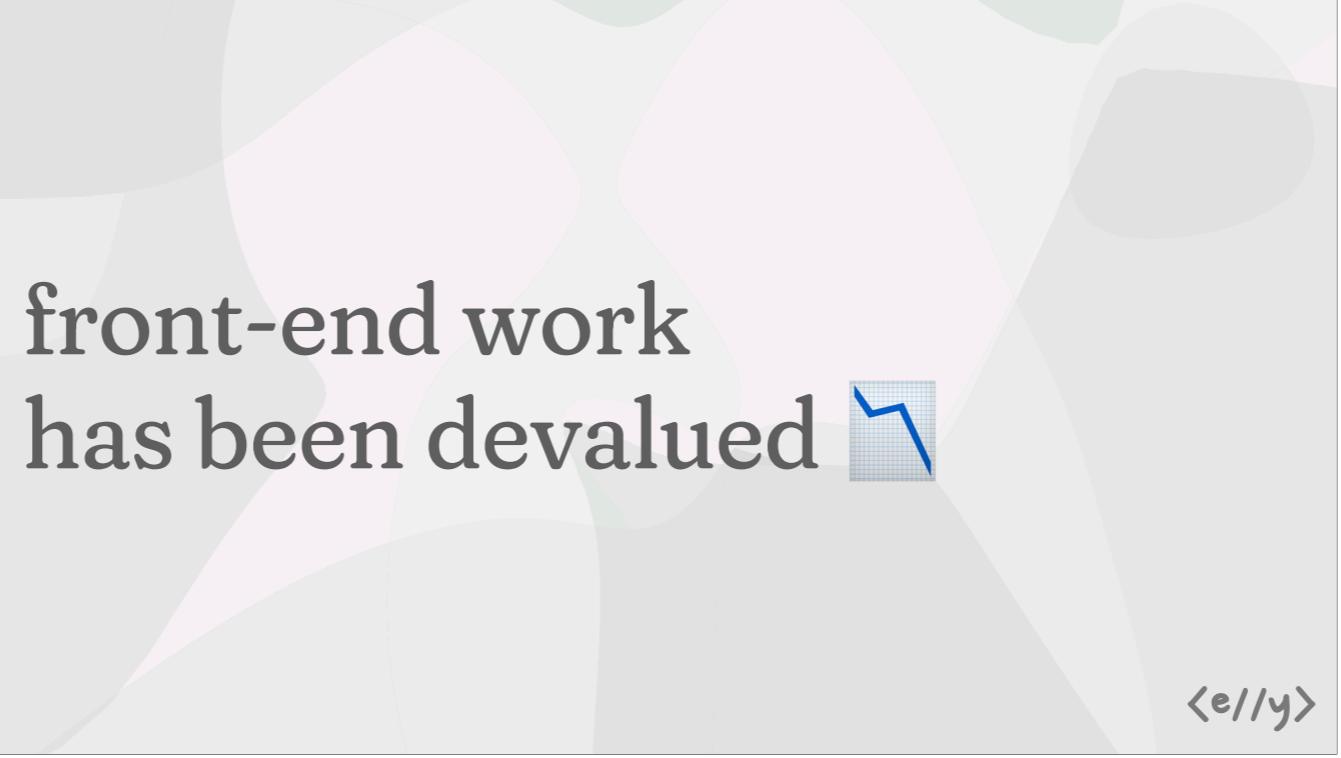
`<e//y>`



*Yikes! I need some
separation of my concerns.*

<e//y>

Yikes! I need some separation of my concerns.

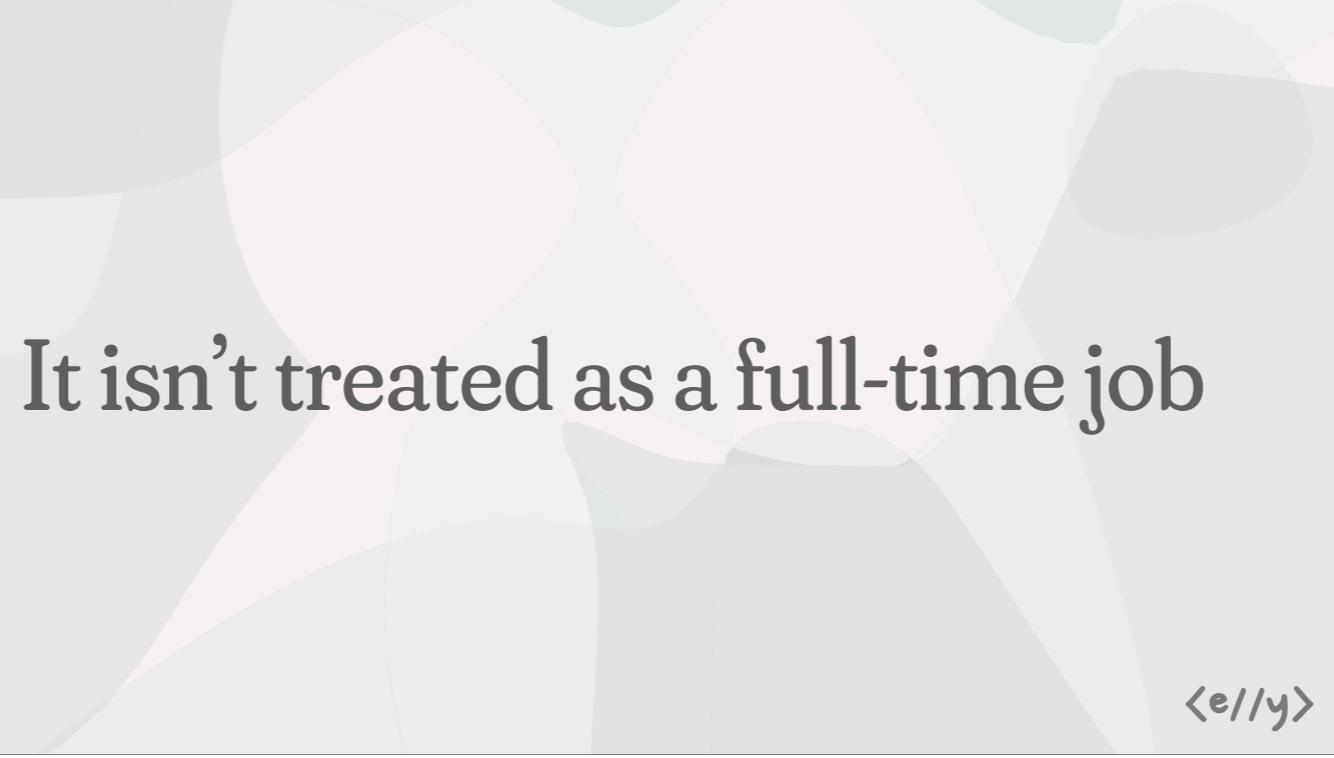


front-end work
has been devalued



<elliy>

I went a bit off topic here but what I'm getting at is this.
Actual front-end work has been devalued.



It isn't treated as a full-time job

<e//y>

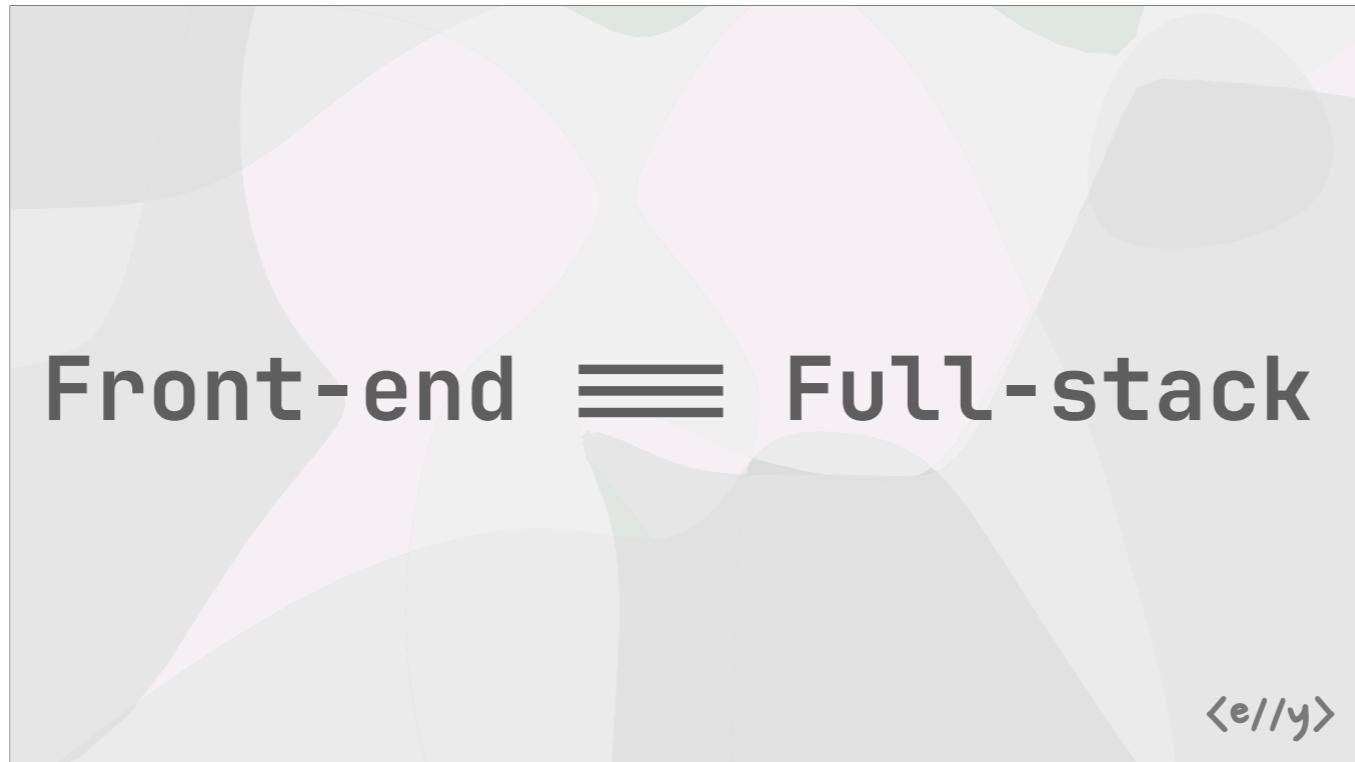
It isn't treated as enough to be a full-time job.



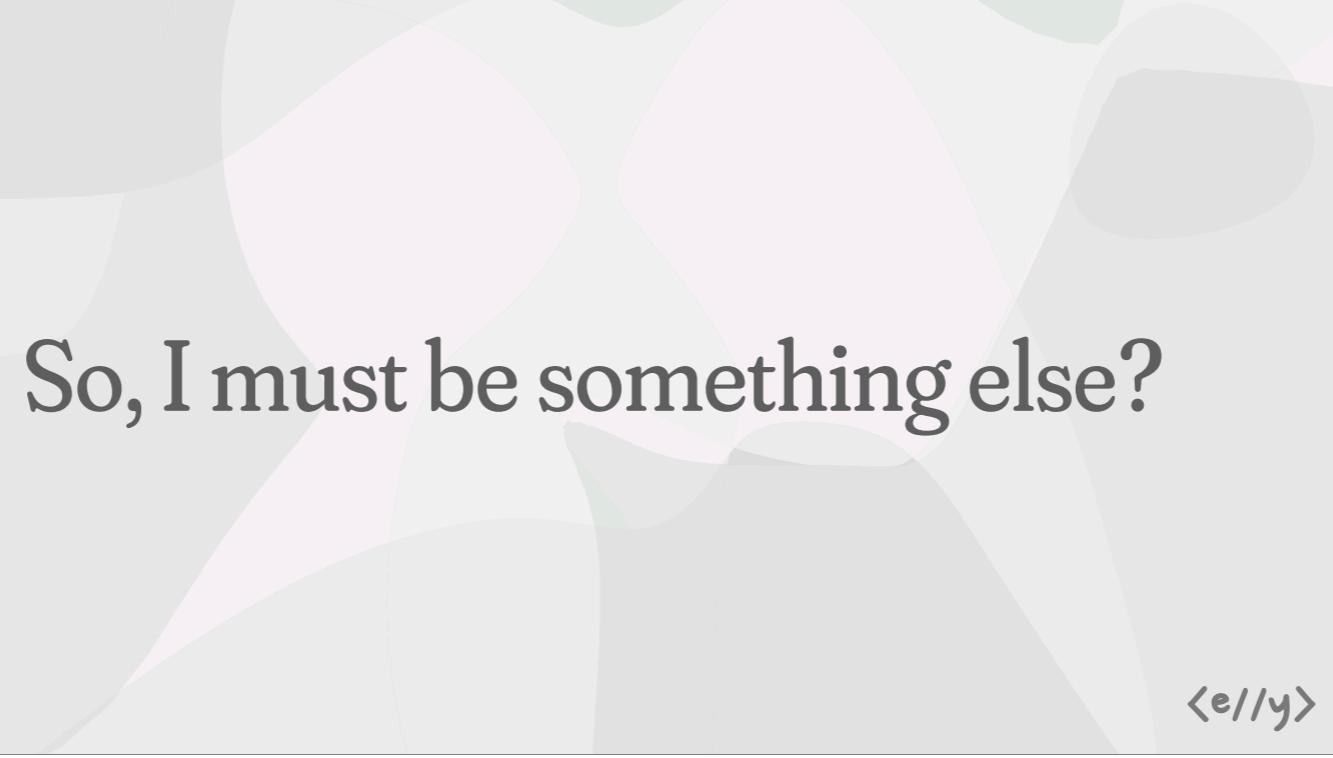
Now front-end no longer
actually means front-end

`<e//y>`

It's gotten to the point that now front-end no longer actually means front-end.



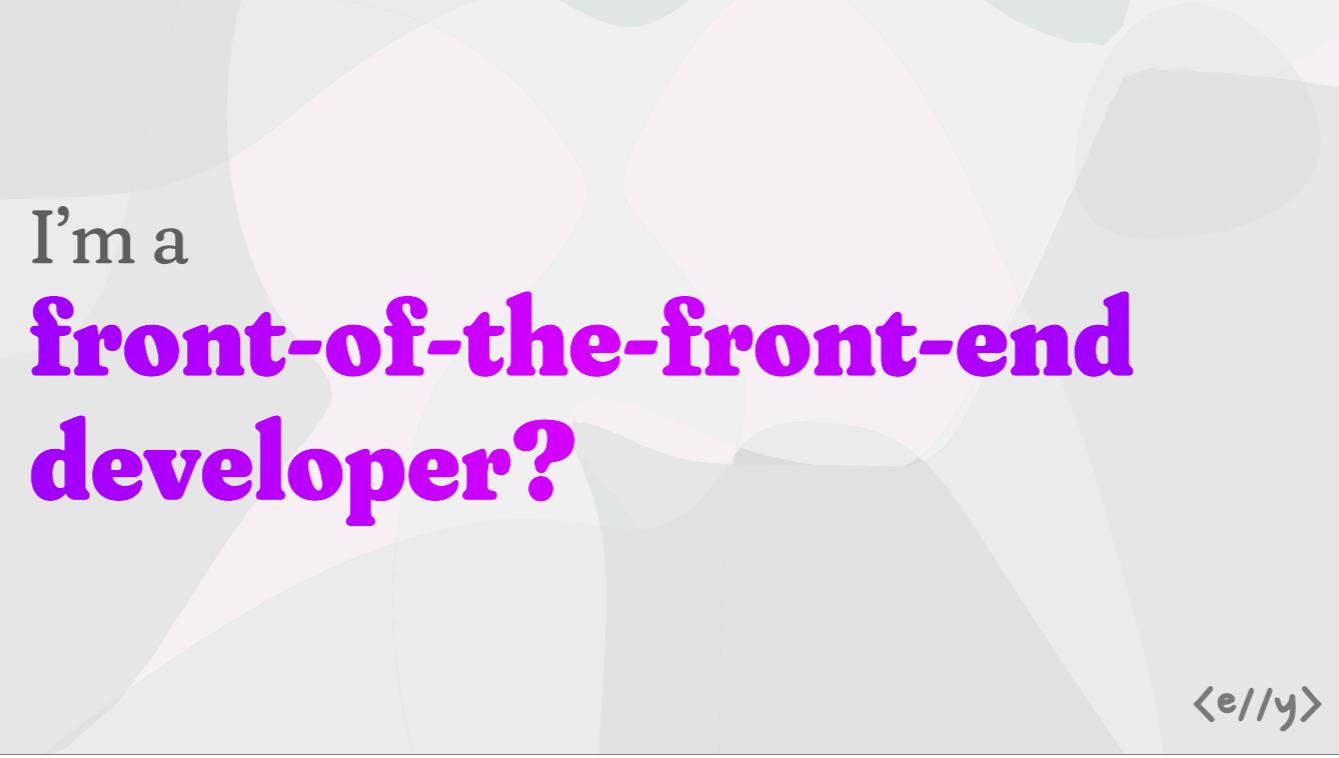
The duties of a front-end developer are more accurately described as full-stack.
And as we learnt earlier, there are many issues with full-stack development as a concept.
I don't want to be a full-stack developer.



So, I must be something else?

<e//y>

So, I must be something else?



I'm a
**front-of-the-front-end
developer?**

<e//y>

Well, according to Brad Frost, I'm a front-of-the-front-end developer...
While those words do accurately describe what I do...

Elly Loel

Front-of-the-front-end De...

Melbourne, Victoria, Australia

1K followers - 500+ connections

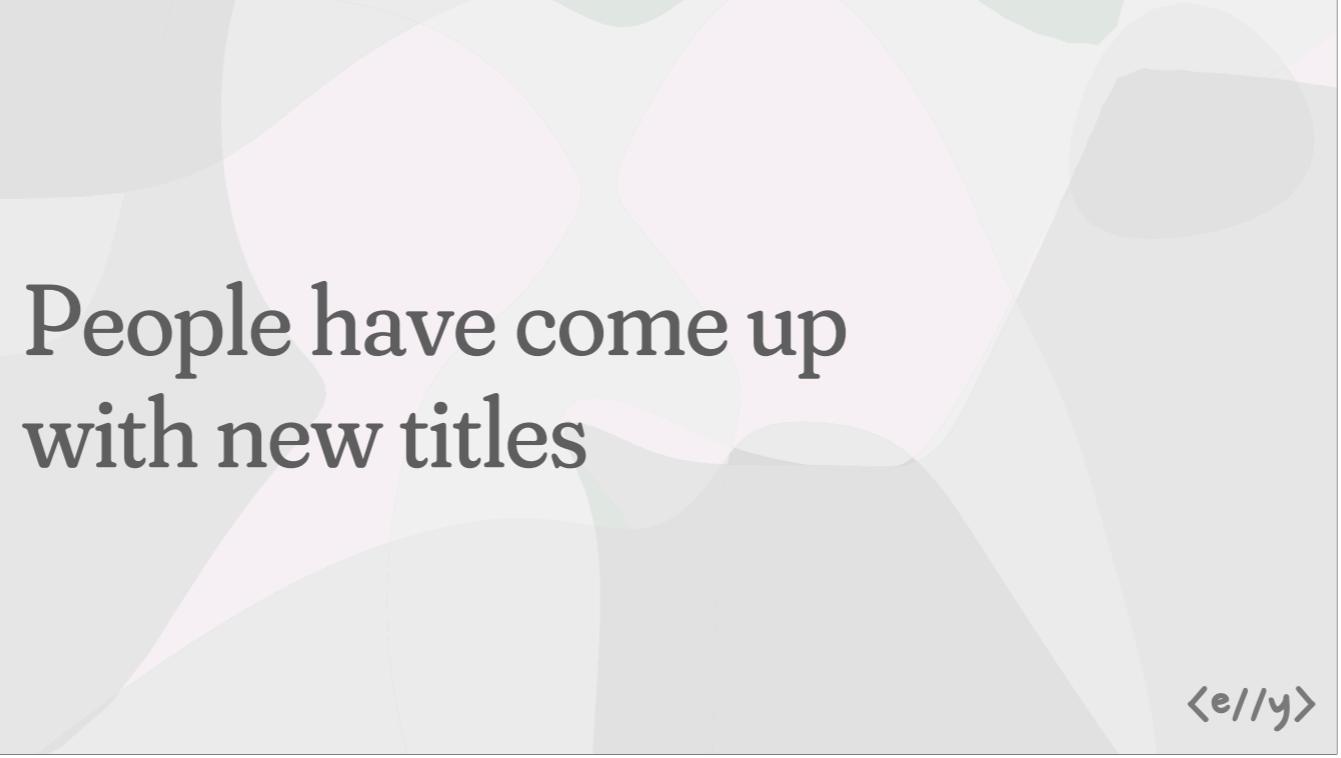
Department of Employment and Workplace Relations

Swinburne University of Technology

They're way too long.

No-one is actually going to use that as their title.

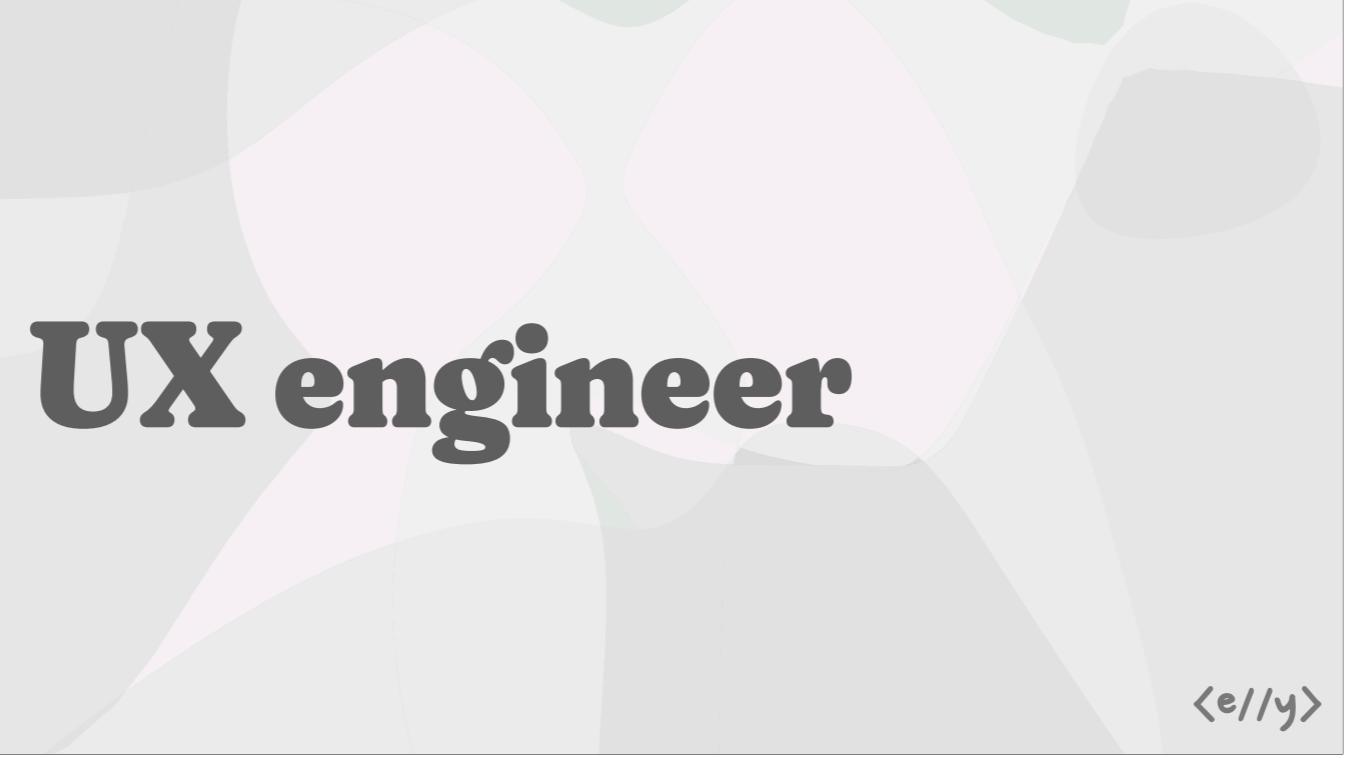
So we need something shorter that still conveys the same meaning.



People have come up
with new titles

<e//y>

People have come up with new titles, like:



UX engineer

<e//y>

UX engineer

Design engineer

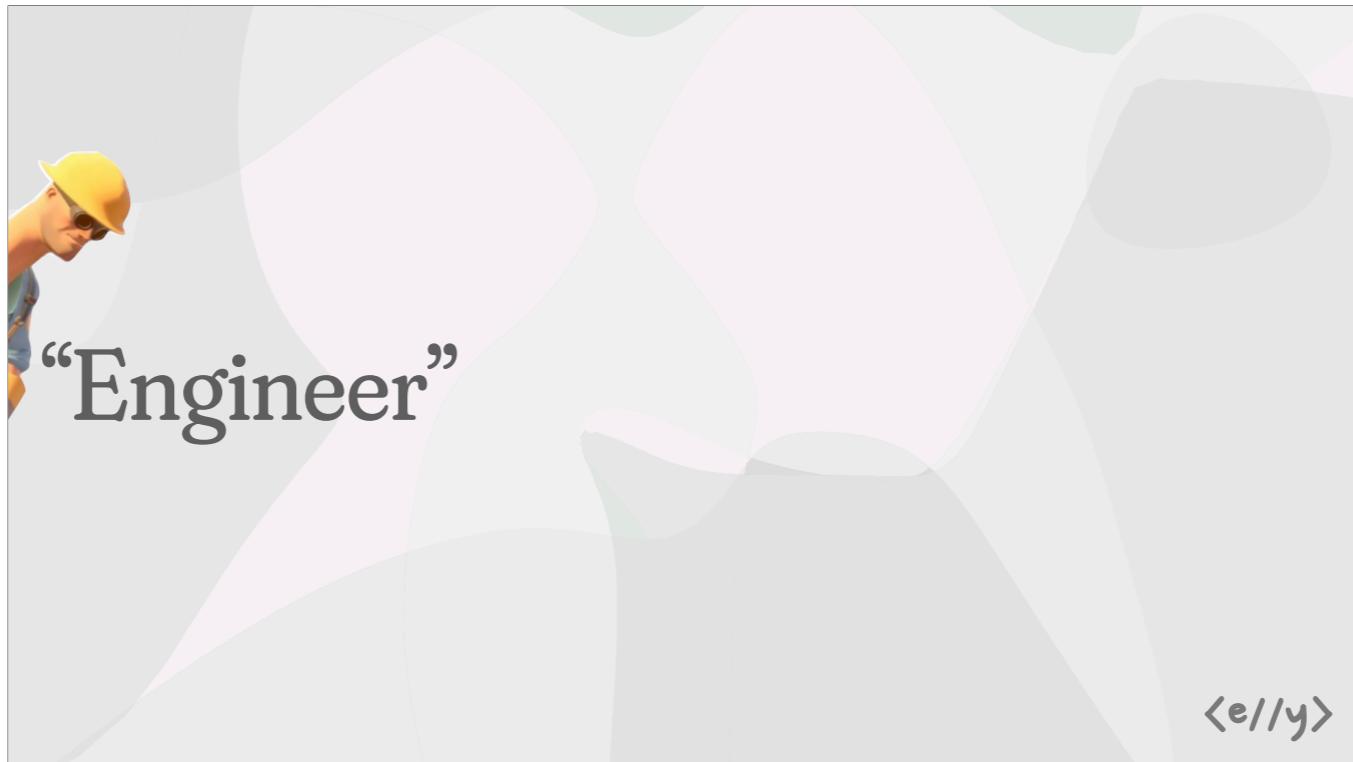
<e//y>

Design engineer



And, I've actually even seen some companies using them!

Though... I've also seen companies *misusing* them, for “front-end” (aka full-stack) roles—which undermines the whole point of coming up with these new titles.



Also, there seems to be a bit of an obsession with having “engineer” in the title.

I’d say that mostly stems from pay related anxiety.

But it’s probably also somewhat tied up in the perception that anything that’s not engineering is not as important.



No.

You shouldn't shove engineer into your title so you get paid more.

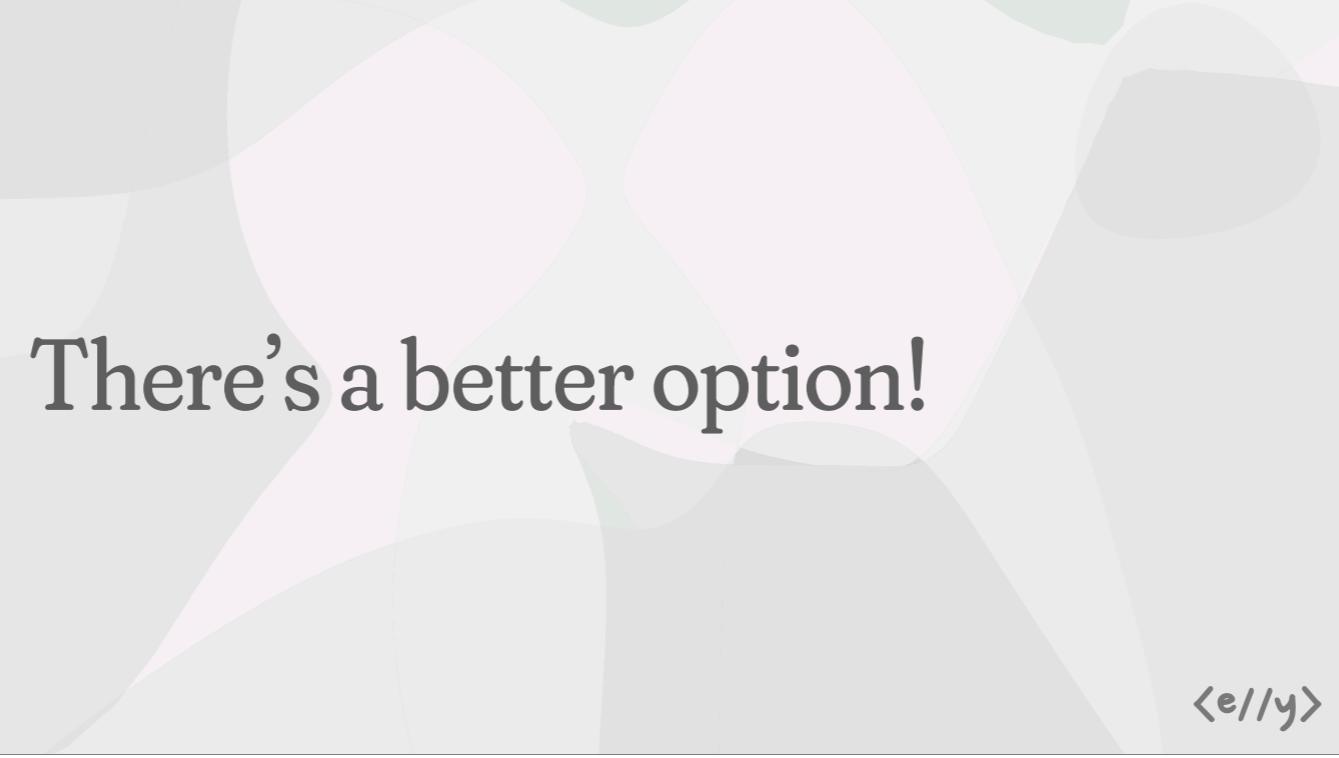
You should help make sure everyone without engineer in their title gets paid more.

We all deserve more.

Web Mistress 
Web Witch 
Web Crafter 
Web Artisan 
Web Worker 
Web Specialist 

`<e//y>`

Look! Here are some titles that don't include engineer.



There's a better option!

<e//y>

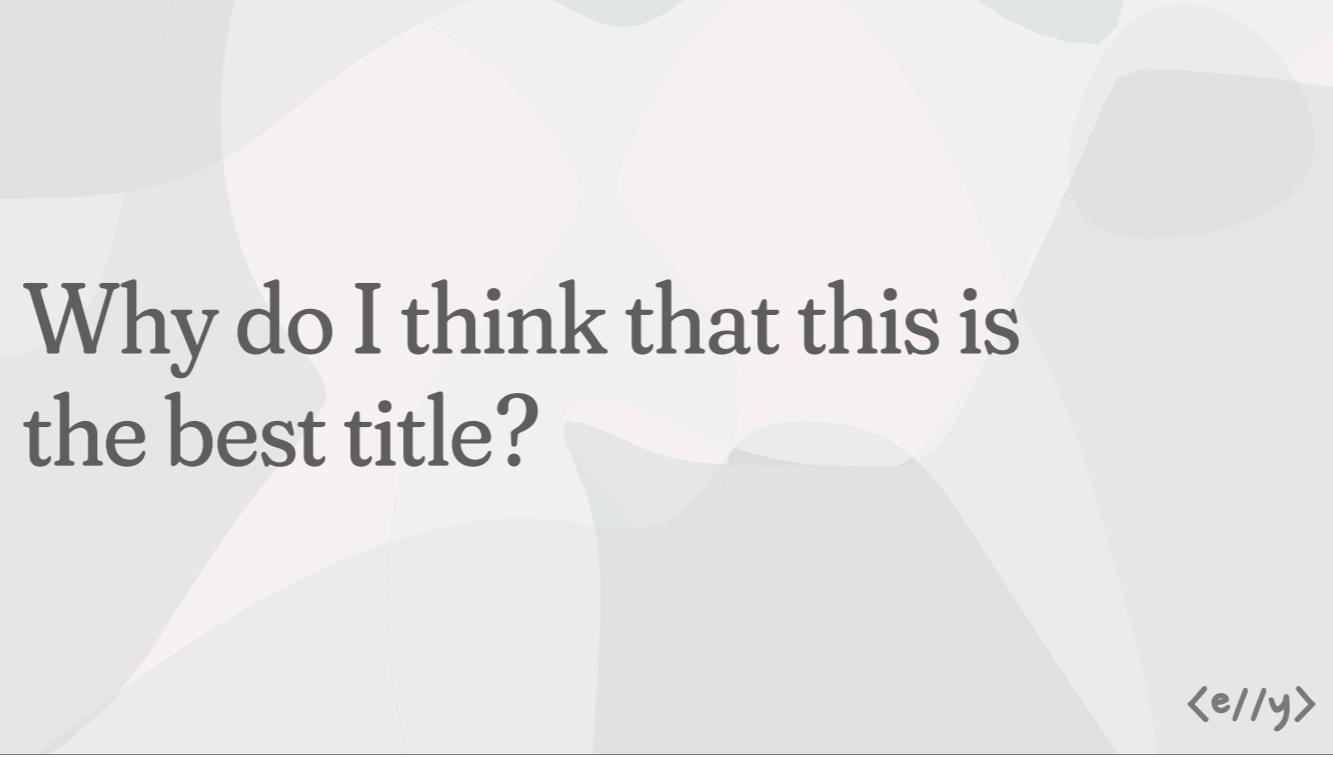
I don't hate the other titles; for a while there I was even using design engineer to describe myself.
But, there's a better option that's been around since long before even "Front-end developer" was being used.
We had the right title in front of us all along...



I'm a
web designer

<e//y>

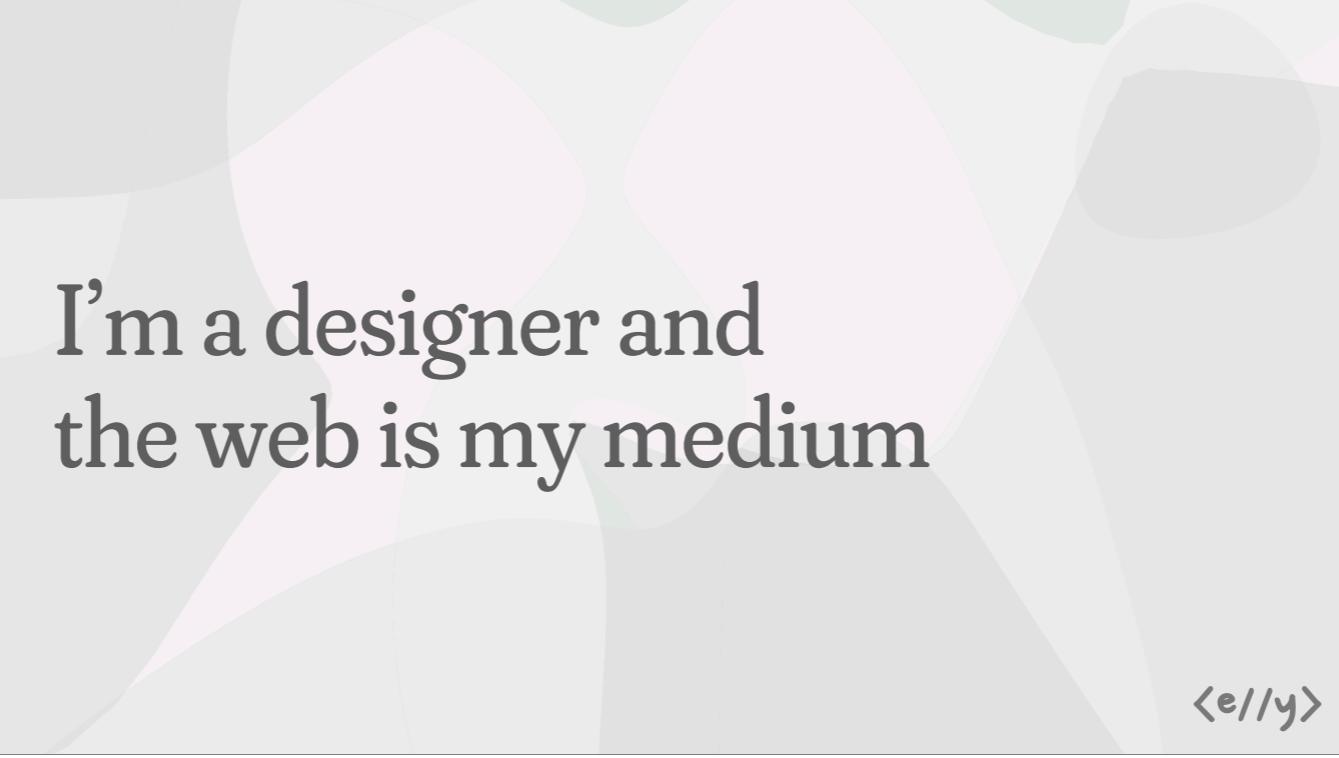
I'm a web designer.



Why do I think that this is
the best title?

<ell>

Okay, so why do I think that web designer is the best title?



I'm a designer and
the web is my medium

<e//y>

Because, I'm a designer and the web is my medium.



I'm designing **for** the web

<e//y>

I'm designing for the web.

The infinitely flexible web.

The web that doesn't have one screen size, one browser, one operating system, or one device.

The web that can be used by anyone, anywhere, on any internet connection, on any device, on any operating system, on any browser, with any screen size.



I'm designing **with** the web

`<e//y>`

I'm designing with the web.

Using the web platform (HTML, CSS, JS, ARIA, etc.), not a bloated harmful abstraction.

I have a deep understanding of HTML and its semantics.

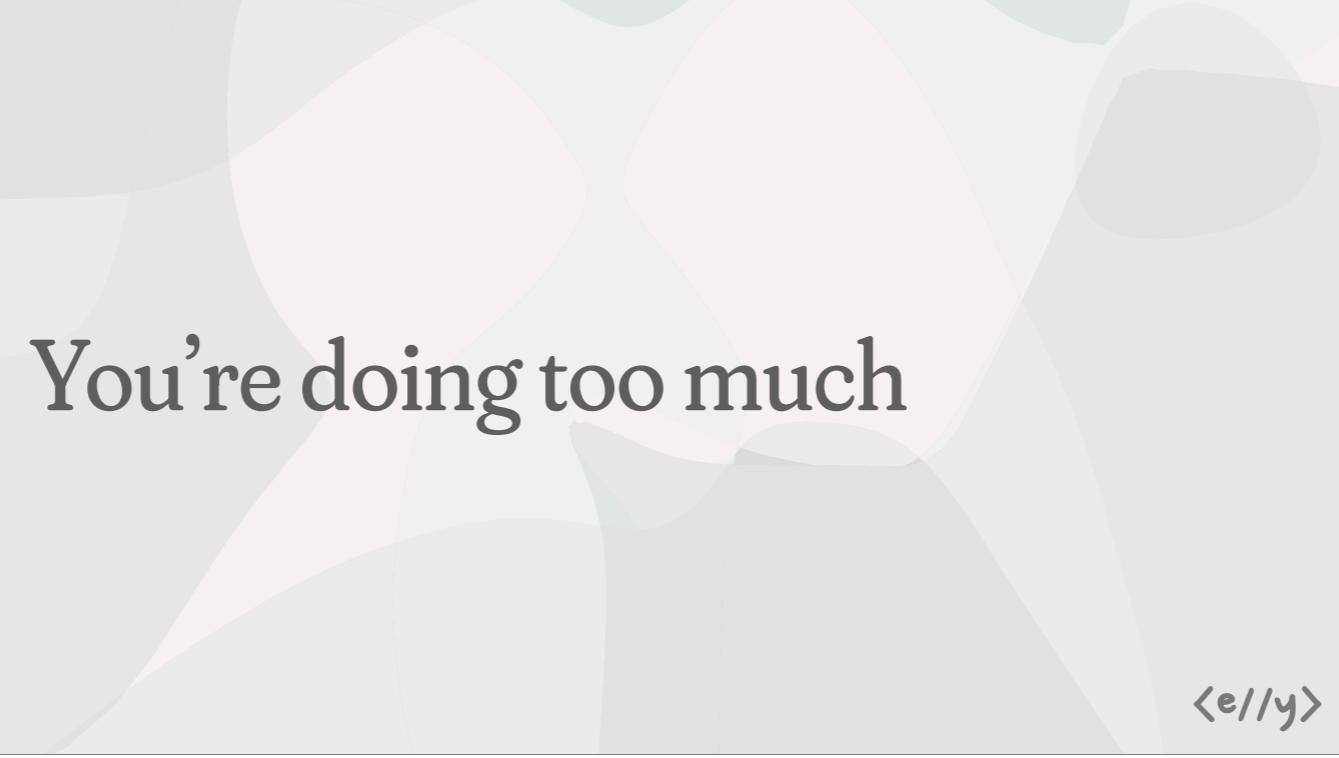
I love CSS, I know how and when to utilise its many features, and I keep up-to-date as more are added.

I have a strong understanding of modern JavaScript and most importantly I know when not to use it.

Why does this matter?

<ell>

Why does this matter?



You're doing too much

<e//y>

In the current state of our industry we're being forced to do too many things. We're spread too thin. It's completely unsustainable which is why we're constantly burning out.

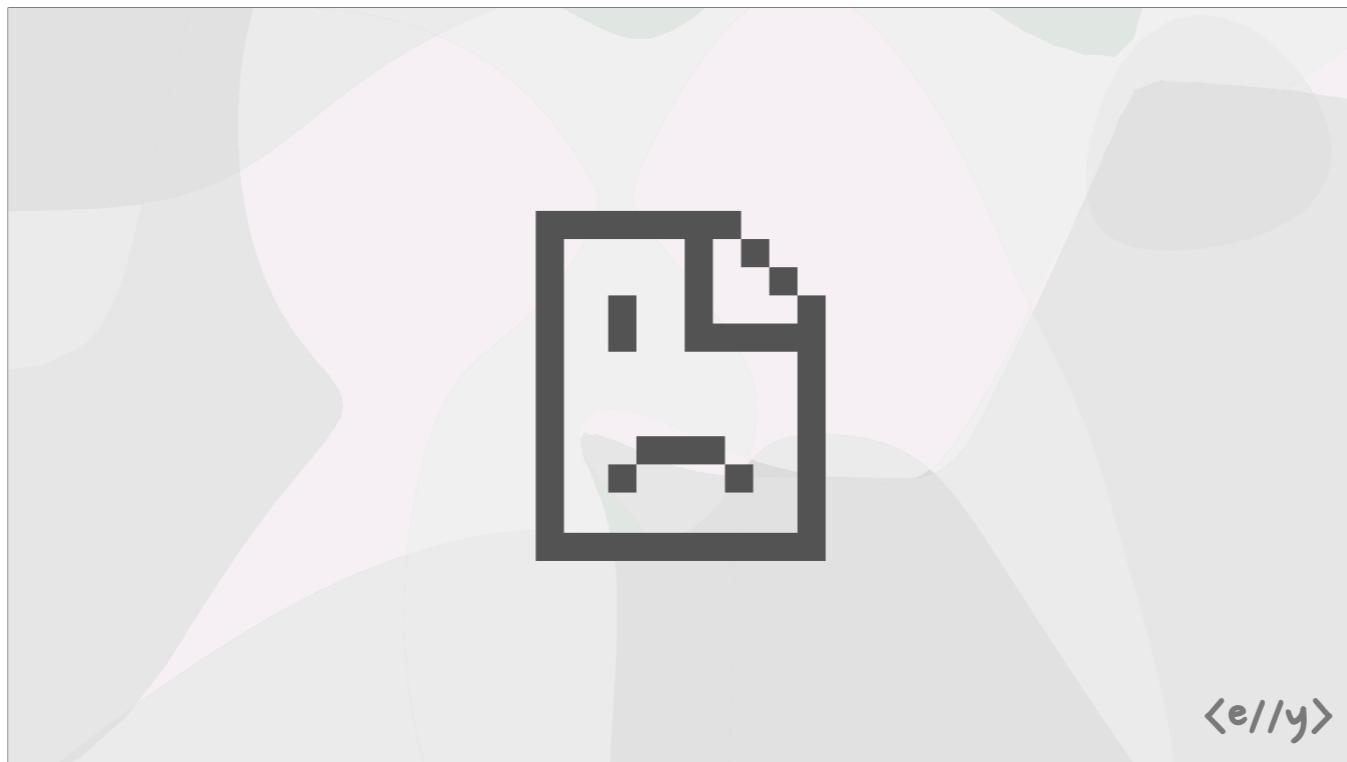


Everyone else deserves better

`<e//y>`

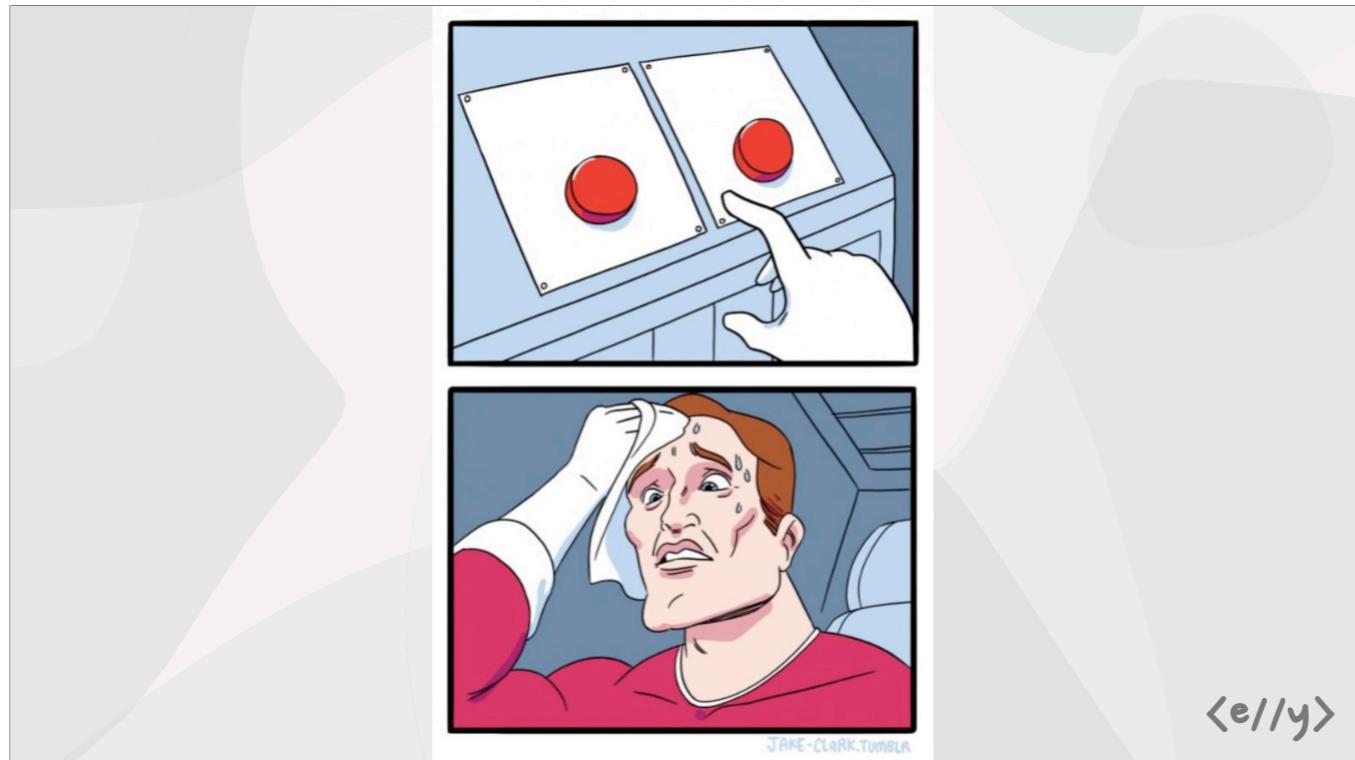
We're forced to do the wrong things which leads to bad outcomes and that's not okay.

Bad code is often some or all of the following:



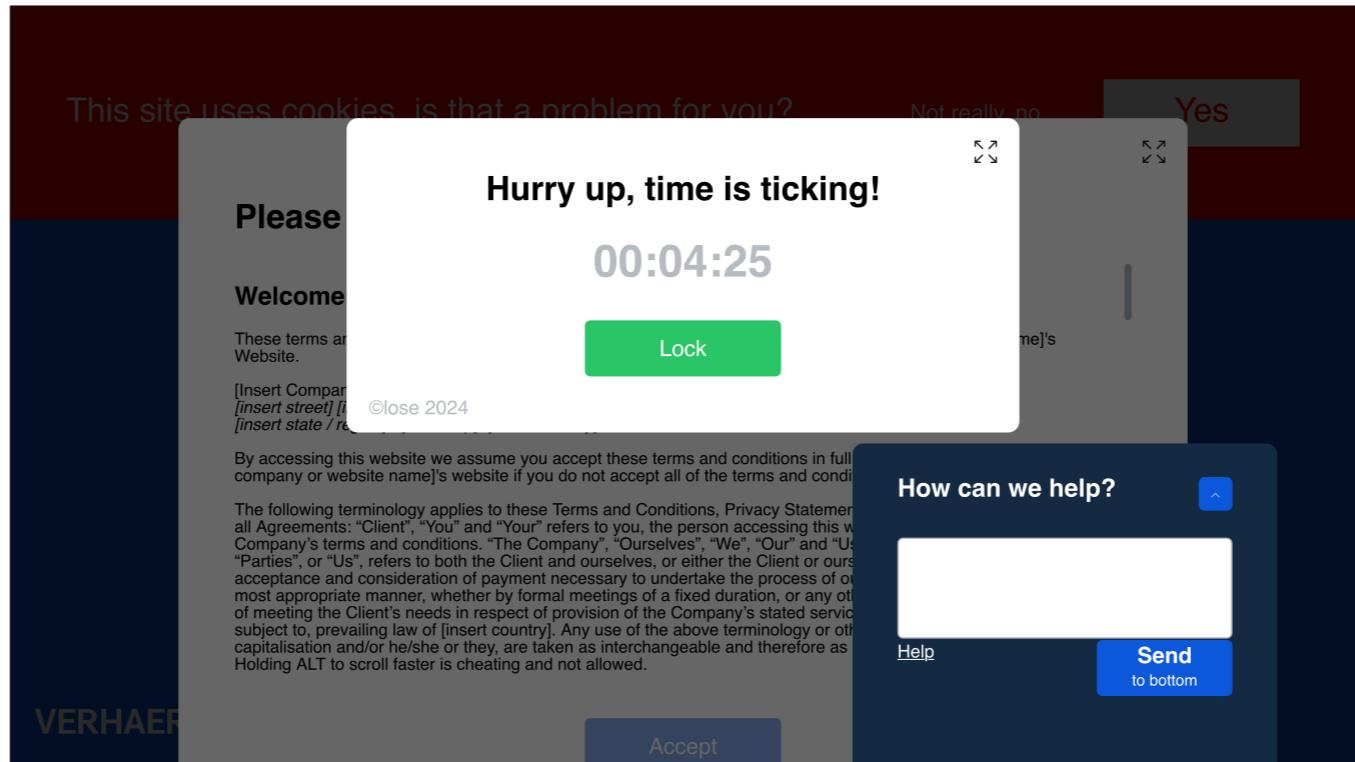
`<e//y>`

Non-functional, so it doesn't work at all.

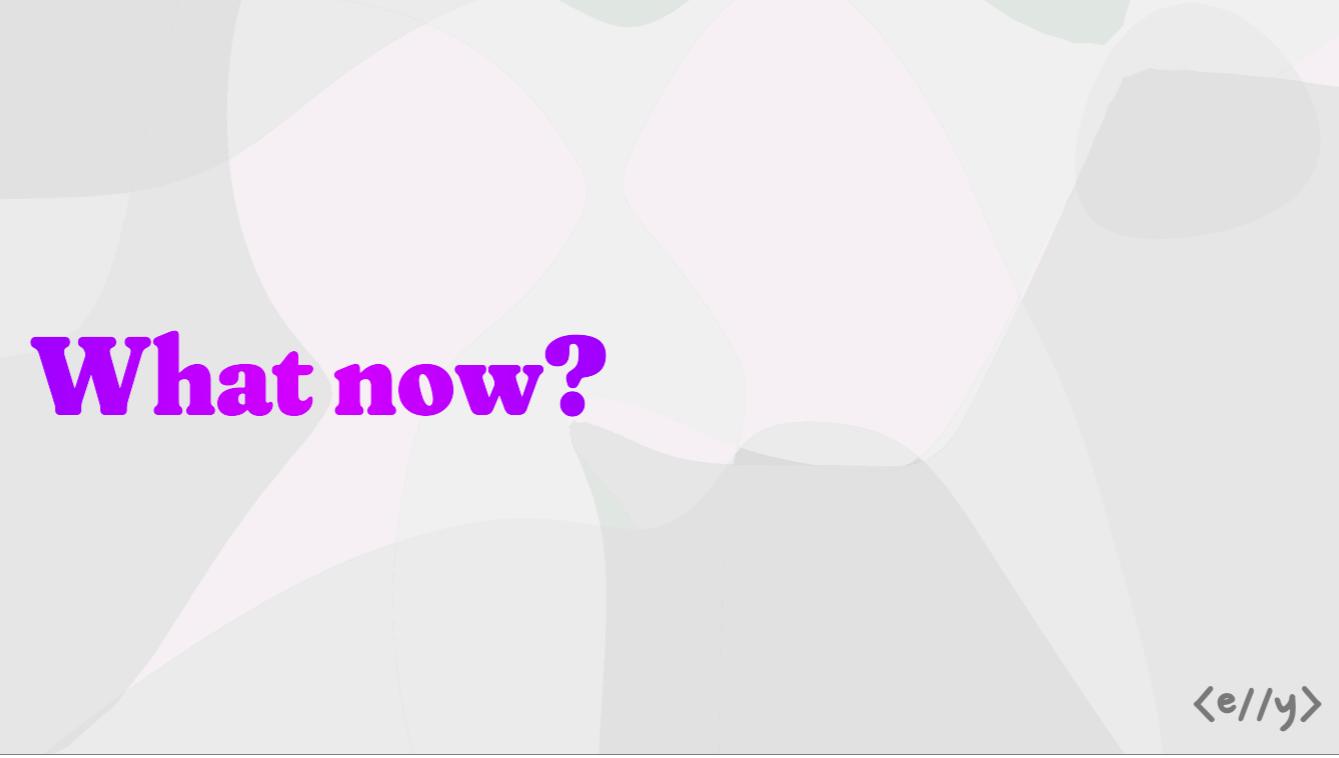


<e//y>

Inoperable/Inaccessible, so it can't be used by some or all people.



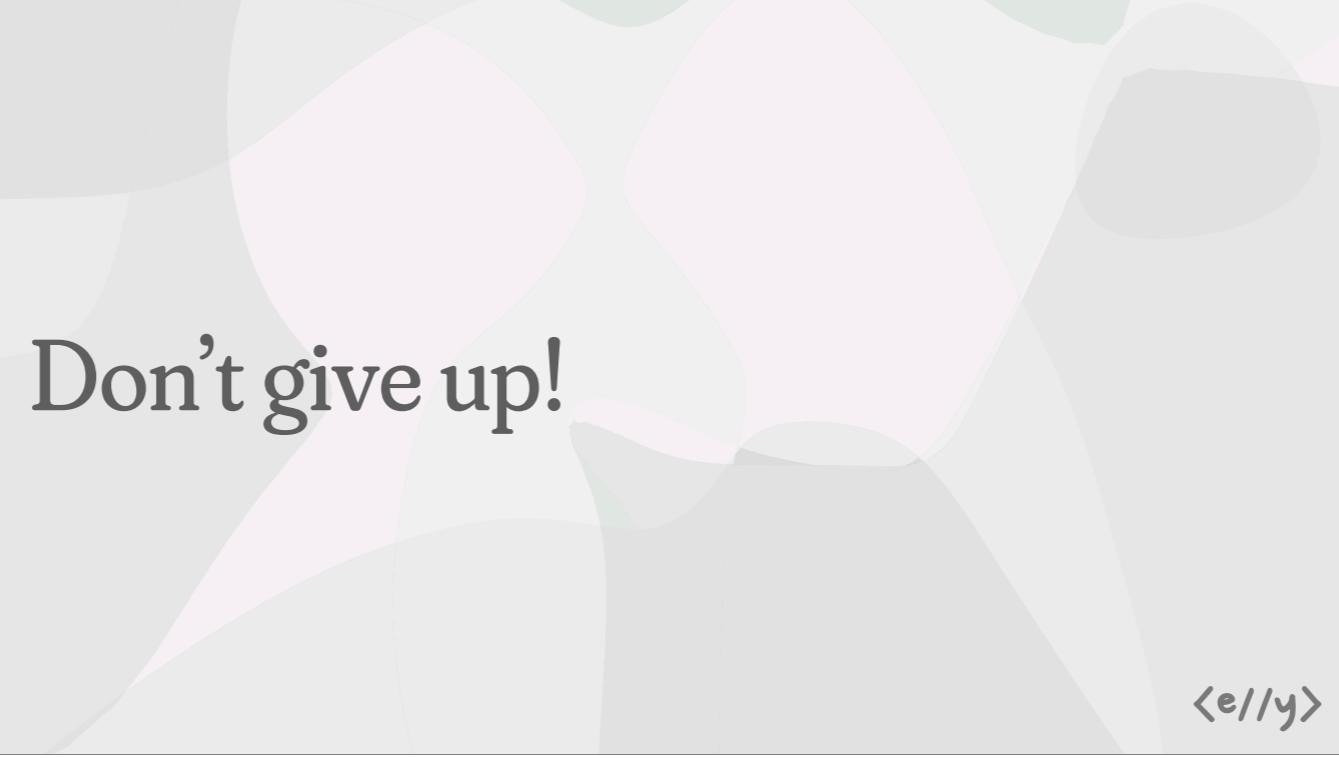
Or unusable, so it's too hard to be used by some or all people.



What now?

<e//y>

Okay, so... what now?



Don't give up!

<e//y>

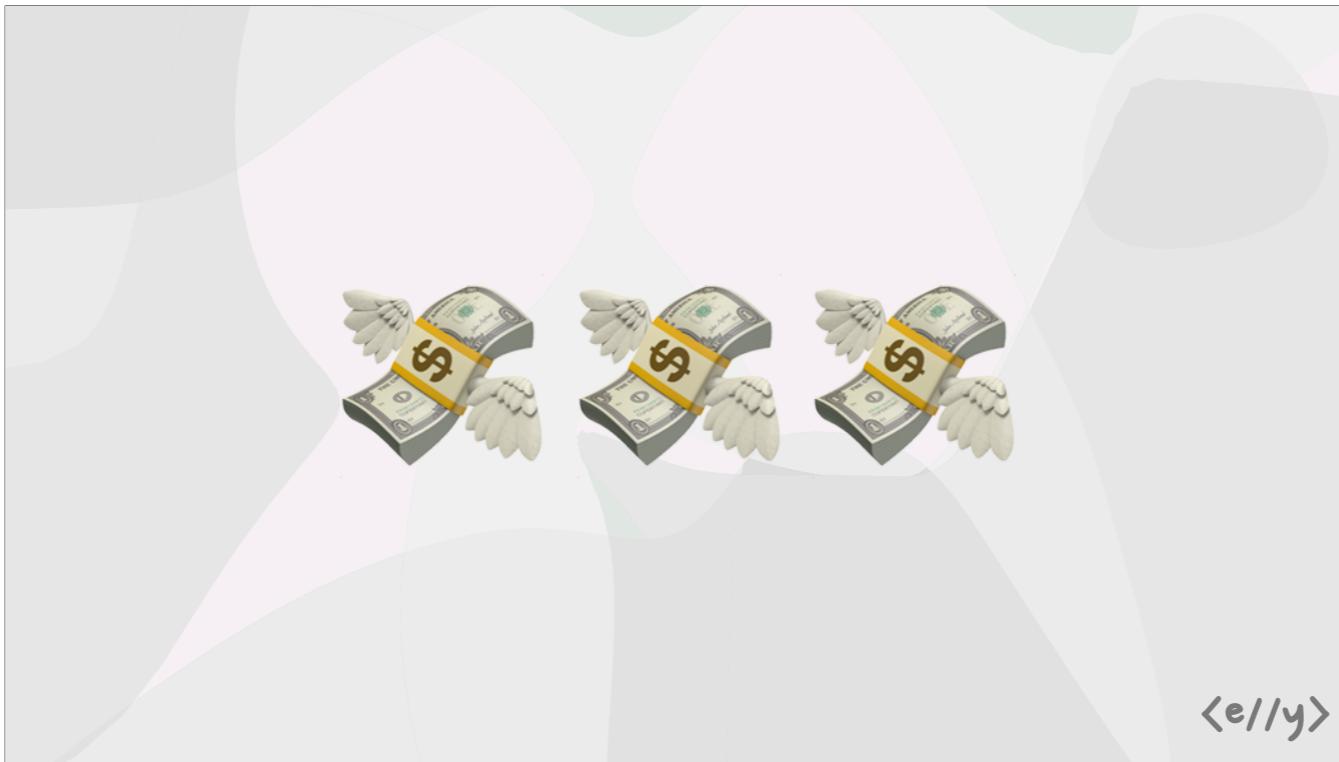
First and foremost, don't give up!

Even through the current landscape of front-end development presents challenges for us web designers we owe it to the people using our sites to not give in to the status quo.



<e//y>

You deserve better than being continuously overworked to the point of burnout before being unceremoniously laid off by some short sighted greedy CEO.
So, you should go and read "You deserve a tech union" by Ethan Marcotte.
Then, join a union and fuck capitalism.



<e//y>

BUT if I were to put my stinky capitalist hat on for a second 🤢

Inaccessible or unusable websites do mean that you're going to have less people using your site.

So, you're going to be making less money than if your site was accessible and usable.

Okay, capitalist hat back off, what else?



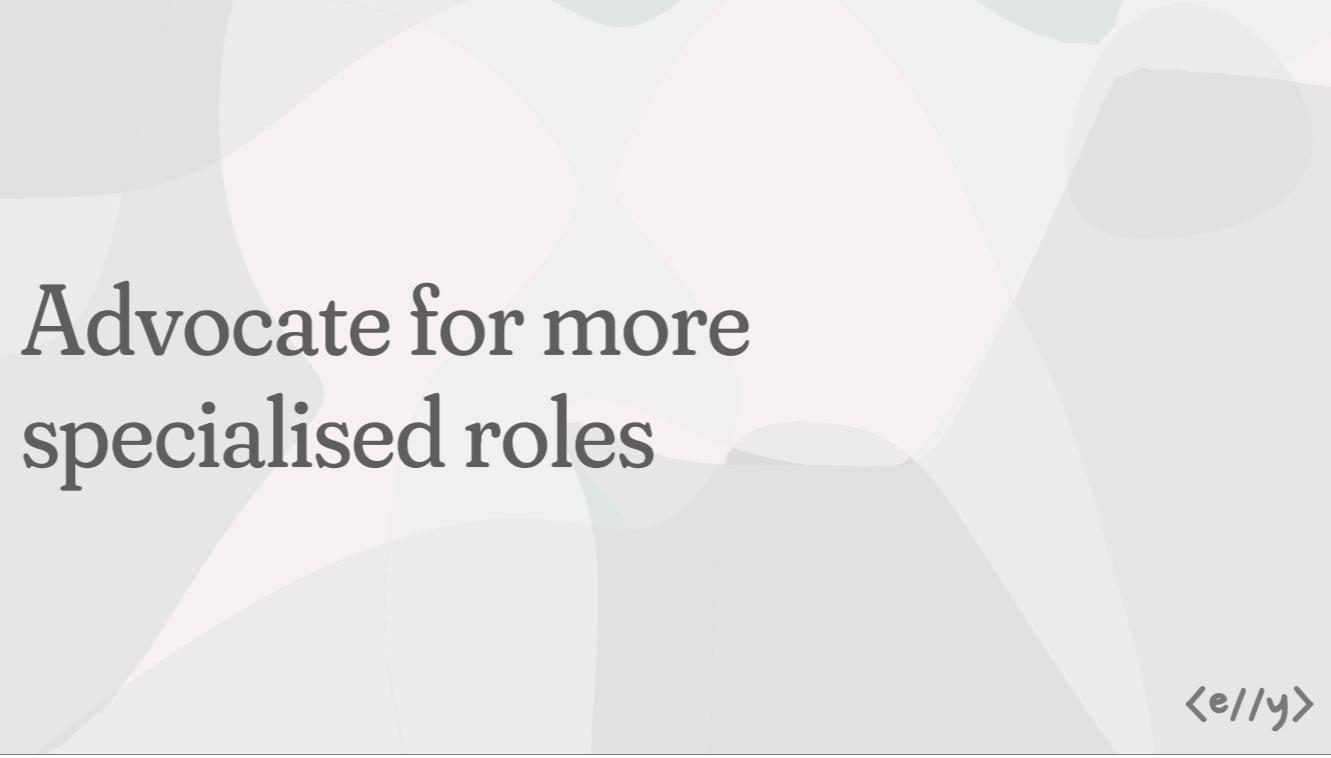
Advocate for preventing overwork

AKA abolish full-stack roles

<e//y>

Advocate for preventing overwork.

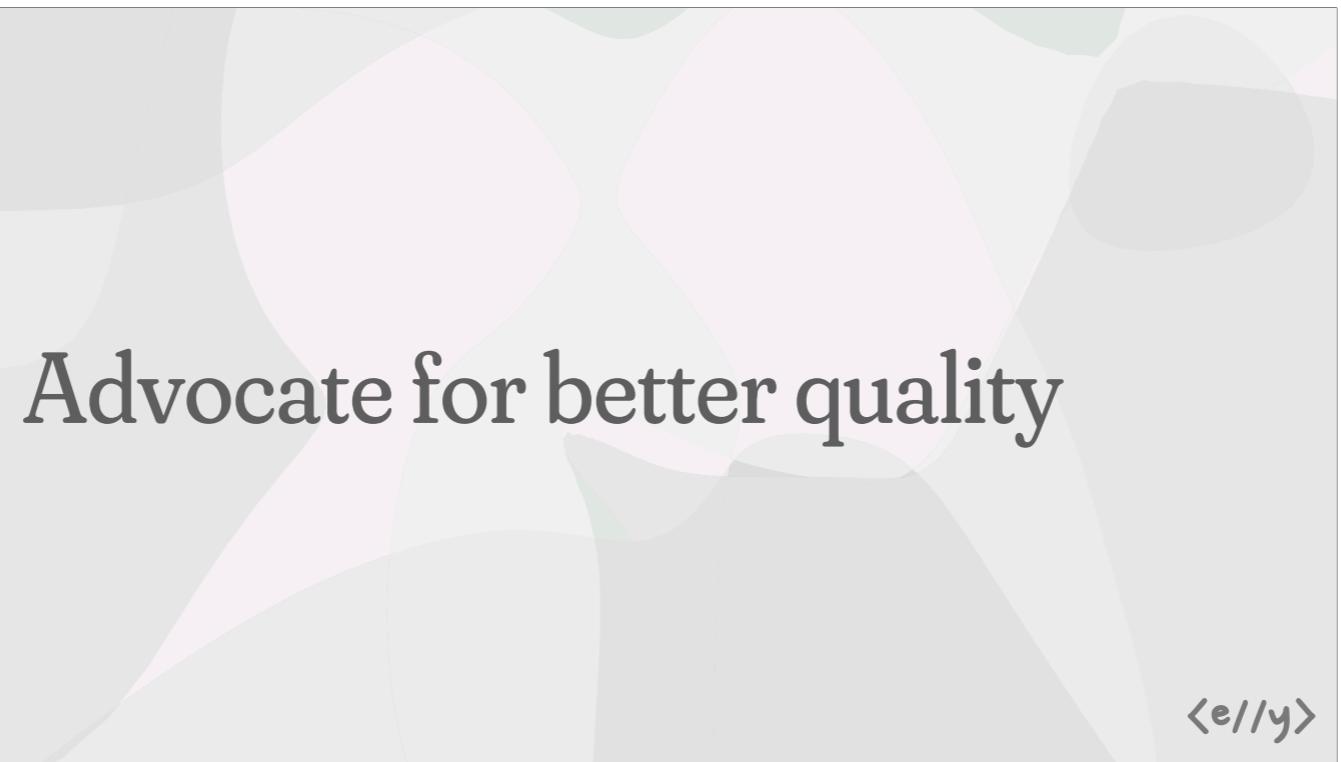
So basically, abolish full-stack roles where you're spread too thin.



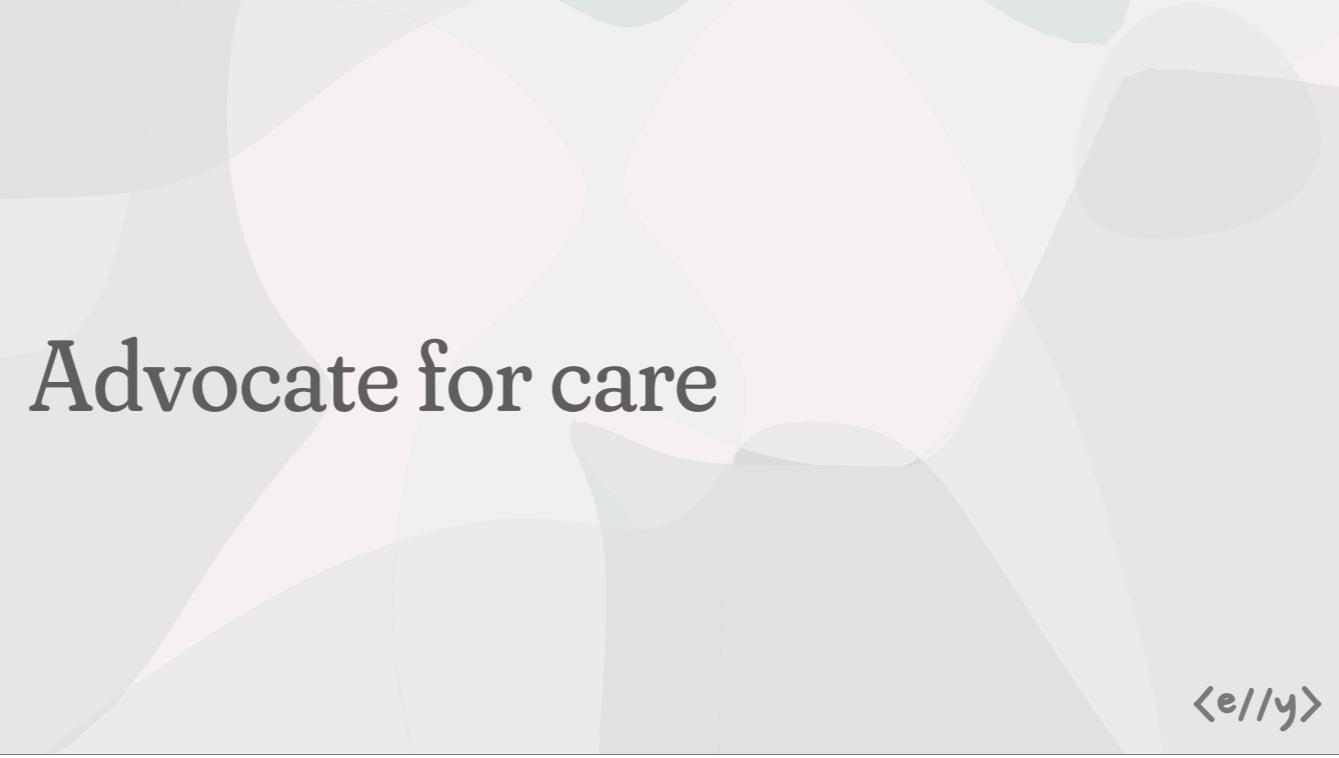
Advocate for more
specialised roles

<e//y>

Advocate for more specialised roles.



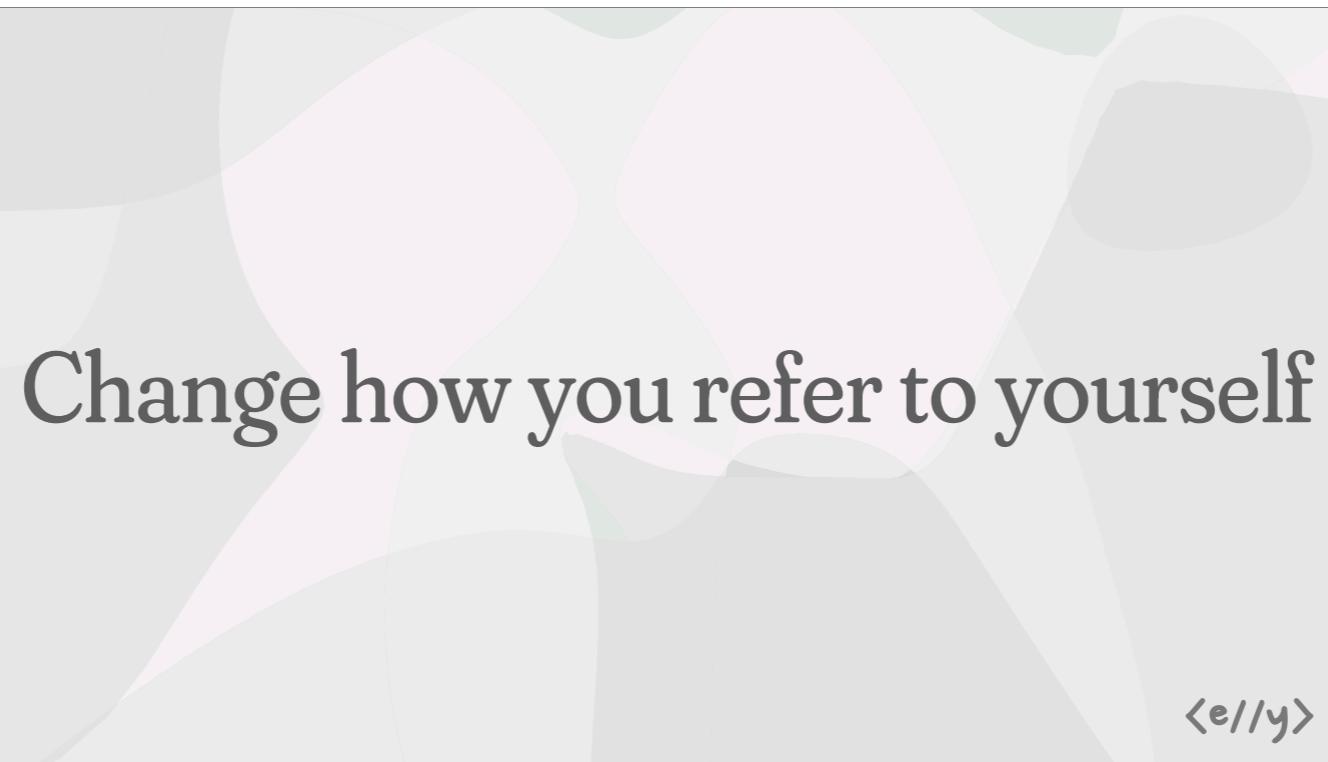
Advocate for better quality.



Advocate for care

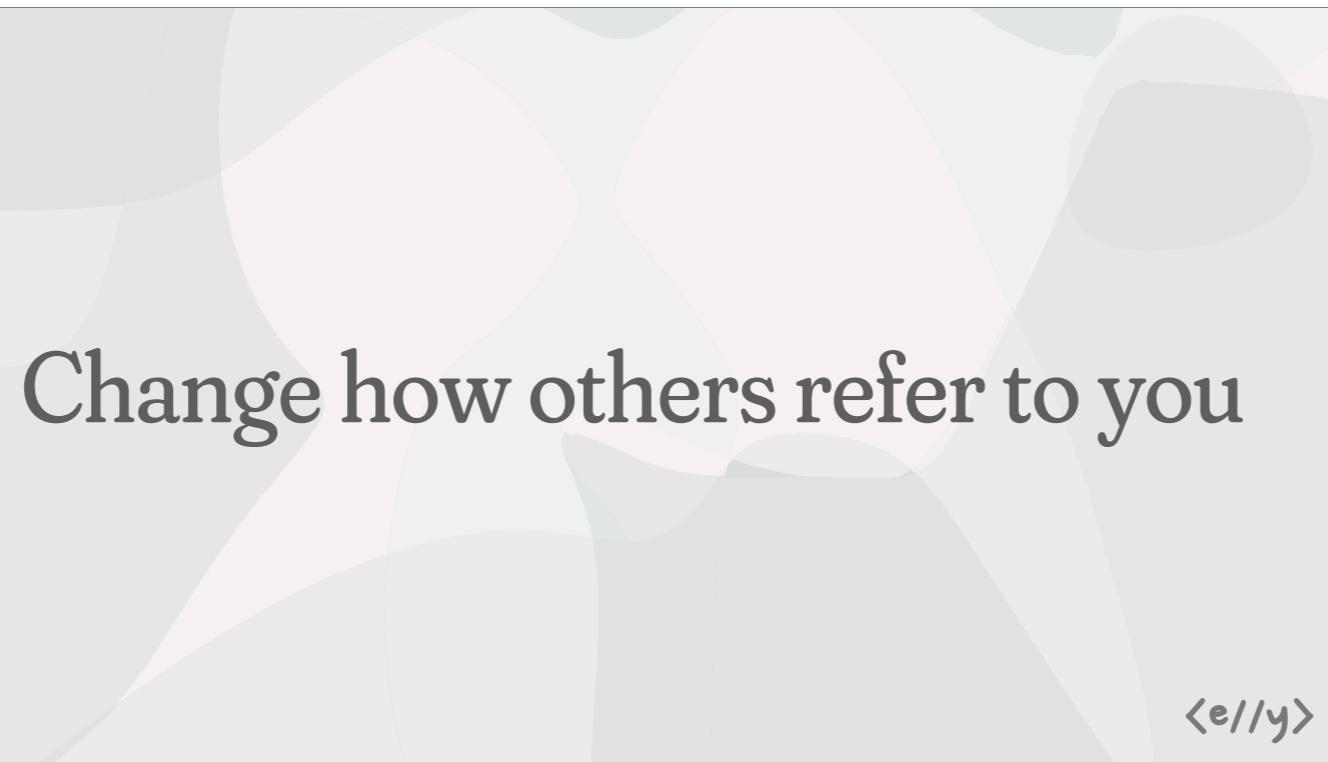
<e//y>

Advocate for care.
Care for our craft.
And most importantly, care for the people using our sites.



Also, we should start changing how we refer to ourselves.

Whether it be when you're talking to co-workers—or even friends and family, on your personal website, on LinkedIn, on your resume and CV, everywhere!



Once you've changed how you're referring to yourself then it'll be much easier to change how others refer to you.

It might not be easy, but it will be *easier*.

Talk to your manager, see if you can change how they refer to you, both officially and unofficially.

If you can get your title changed that's amazing.

If not you should still absolutely be able to get everyone to change how they refer to basically everywhere with the only exception being things like official documents.

Leadership & management

<e//y>

Now I'm talking to the people who make these decisions.

Yes, you! 🤜

You need to make sure this change happens.

You are the ones with the power to improve our current situation.

If you need to, make any and all changes that are needed to make this change happen.

Do whatever you can!

- Change the role titles.
- Change the team or department names.
- Update your job ads so titles and descriptions are accurate.

And most importantly, if someone is in fact doing 2 or 3 jobs give them 2 or 3 titles and pay them 2 or 3 salaries!

Thanks for listening!

🔗 ellyloel.com

🐘 @elly@front-end.social



Thank you so much for listening everyone!

You can scan that QR code or head to the link ellyloel.com to head to my website where you can also find links to all my other socials.