I'm Jack Harrhy

And this, is

why the web

a big thanks!

why the web

why the web?

why the web?!?!?!

similar to a previously delivered talk, but
geared to a more intermediate / senior audience

why the web

some quick questions for the room!

working at a company
that has some web
presence
/
built websites previously

considers themselves
a "full-stack developer"

writes JavaScript/TypeScript
on a regular/semi-regular basis

enjoys writing CSS

for those that didn't
raise their hand, i hope this
is still an interesting talk to
listen to,

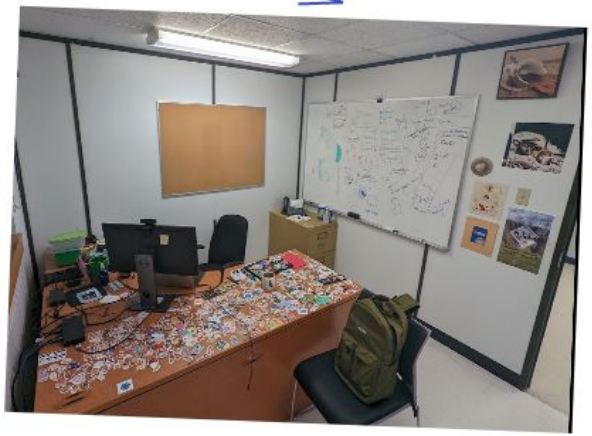
for those that did, hopefully
what i have to talk about here
might be useful to you

:)

Obligatory whoami slide:

My background:

my sticker-clad office :)



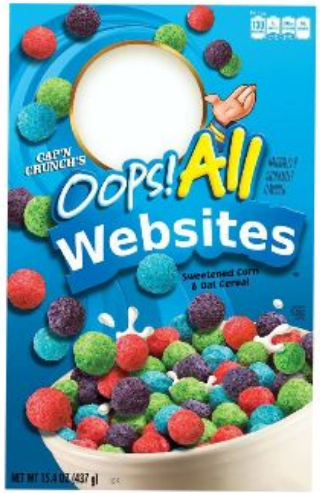
Currently I work at C-CORE: Engineering Stuff

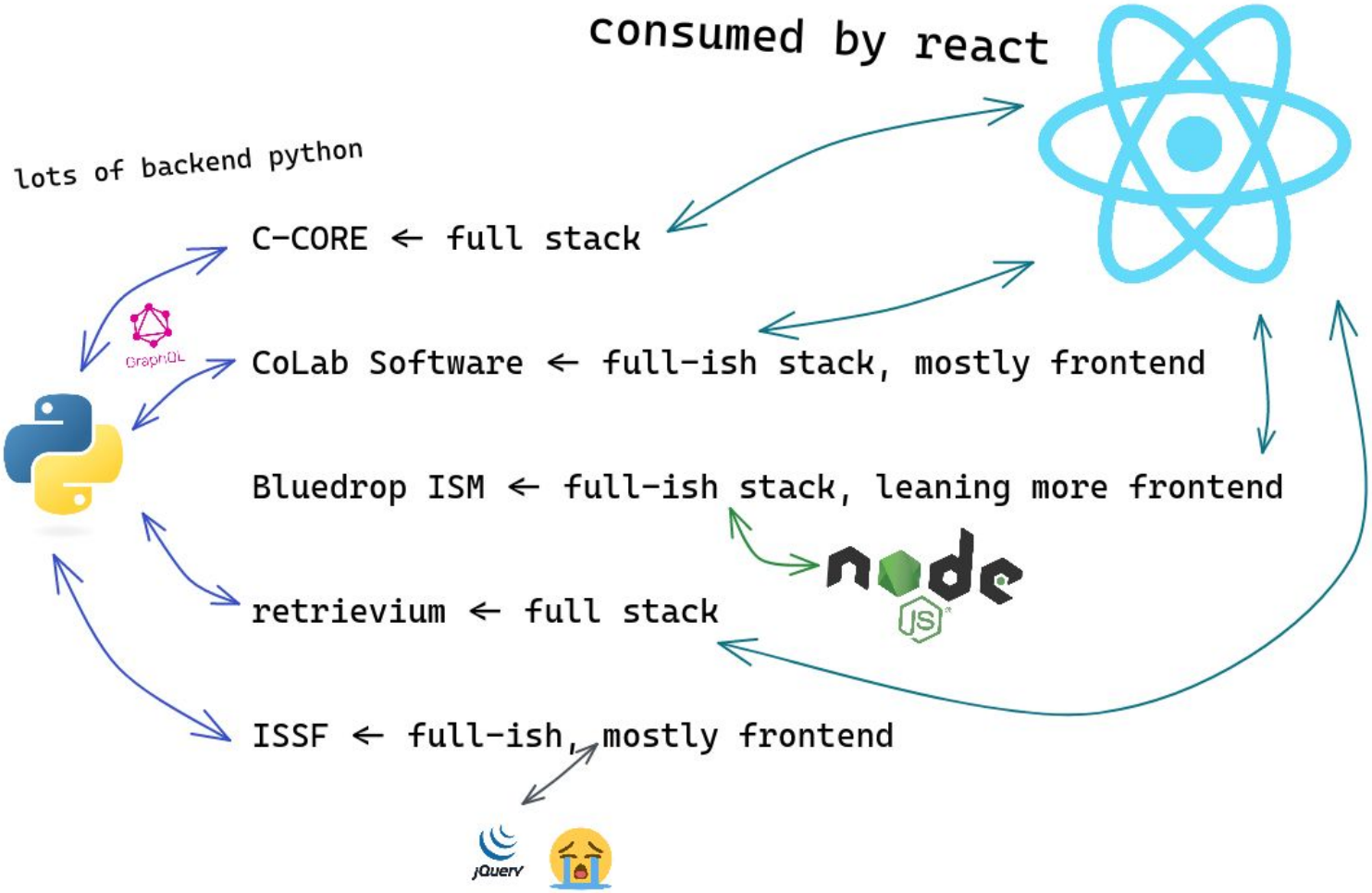
Previously I've worked for:
CoLab Software: GitHub for Engineers
Bluedrop ISM: Integrated Skills Management

And worked on some random projects:
retrievium - discovery tool for chemistry
ISSF - Information System on Small-scale Fisheries

C-CORE
CoLab Software
Bluedrop ISM
retrievium
ISSF

Websites!




consumed by react
lots of backend python
C-CORE ← full stack
CoLab Software ← full-ish stack, mostly frontend
Bluedrop ISM ← full-ish stack, leaning more frontend
retrievium ← full stack
ISSF ← full-ish, mostly frontend
GraphQL
node
jQuery

the technologies / tips & tricks / etc. i'm about to talk about in this talk, should be taken with a considerable serving of salt

they have been useful in helping me solve problems and build websites, but the same may not always translate to everyone here!

however ...
i am going to spend lots of this talk
talking about what i find useful, whenever
you see a slide formatted like this:

tip: *some amazing insight from
jack about web dev*

remember:

outline

- brief terminology
- problems with the web
- how did we get here?
- things aren't that bad
- tools, technologies, & tips

(brief overview of)

terminology

the "triforce" of web technologies

HTML: Hyper Text Markup Language

CSS: Cascading Style Sheets

JS: JavaScript

its related to java, right?

HTML: the skeleton

the first thing sent from
the server → client

holds everything together,
references the rest of the files

sometimes, usable / viewable
on its own just fine

CSS: the skin/hair/clothes

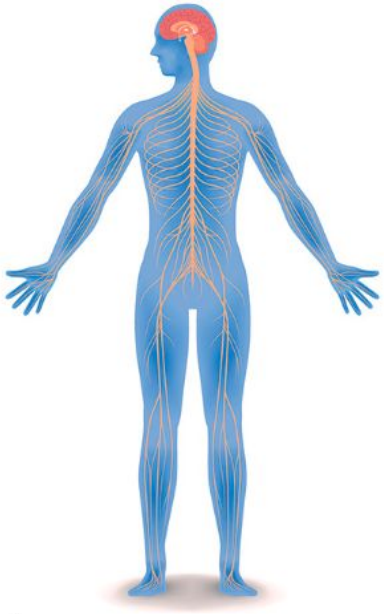
is applied to the html to give it look & feel

not concerned with content, but how to make content not look 'default'

the 'style' of the page 🎨

JavaScript:
the nervous system / brains

takes a static page, and
makes it interactive ✨



permits developers to add
logic to pages, i.e.
- when user clicks button,
  send email
- after the page loads, fetch
  more data from the server,
  such as notifications
- has the ability to modify the
  'DOM' (document object model),
  so it can visually change the site

web browsers: the artist

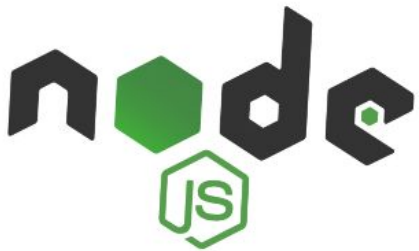
a bundle
of files

we take in html/css/js,
and render an interactive
browser window to the user!

the user,
you!

nodejs / node:
javascript outside the browser

can be used to create
serverside applications
using frameworks like
express



commonly used in frontend
dev tooling as well

npm: node package manager

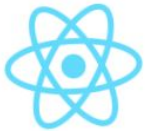
other peoples code, 'packaged' up
and made accessible to other people,
because when building a website you don't
want to reinvent the wheel!

npm cli: what you use on the terminal

npm registry: the default (& defacto) package registry

npm, inc.: a company that oversees the above

vue



react

angular

"the big 3"

frontend(?) frameworks/libraries

svelte

solidjs

"new kids on the block"

remix

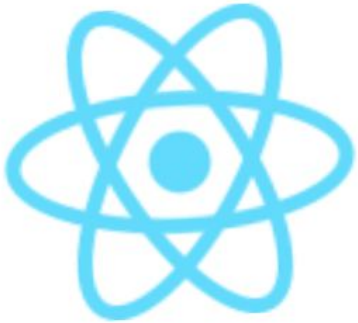
nextjs

"frontend?"

jquery

knockout

"old school"



react

remember:

that was our brief review of terminology, lets jump right into some problems

problems with the web:

build systems

so many browsers to support!
what if i want new features now?

slow connections are
common in some areas!
i can't just ship 50+ files
over a network separately ...

is our 'triforce'
enough to solve these
problems?

that sounds great, right?

an unfortunate increase in complexity

ill take all of the above and produce the output, but oh boy will you need to configure me!

before:

```
my-project/
  index.html
  style.css
  script.js
```

i create a bunch of files,
and while im editing them,
i have the index.html file
open in my browser!

oh yeah, install these too!


```
my-next-gen-project/
  dist/
    ...
  config/
    webpack.common.js
    webpack.dev.js
    webpack.prod.js
  public/
    index.html
  src/
    main.js
    other-stuff.js
    foo-bar-baz.js
    random-stylesheet.css
  package.json
  package-lock.json
```

after build, some ugly mess of files for the browser

webpack will do the heavy lifting!
but gotta setup all the plugins,
and give it definitions for dev & prod!

sort of looks like a html file, but usually
the pre-build html file that webpack processes
with its many plugins

this src/ directory is what we wanted
from the start, but at what cost!

oh yeah, and since we need to install
build tool dependencies, we will need
these files to tell npm what to download too!

That's right, I use React

R I can't

E Setup

A Node environment

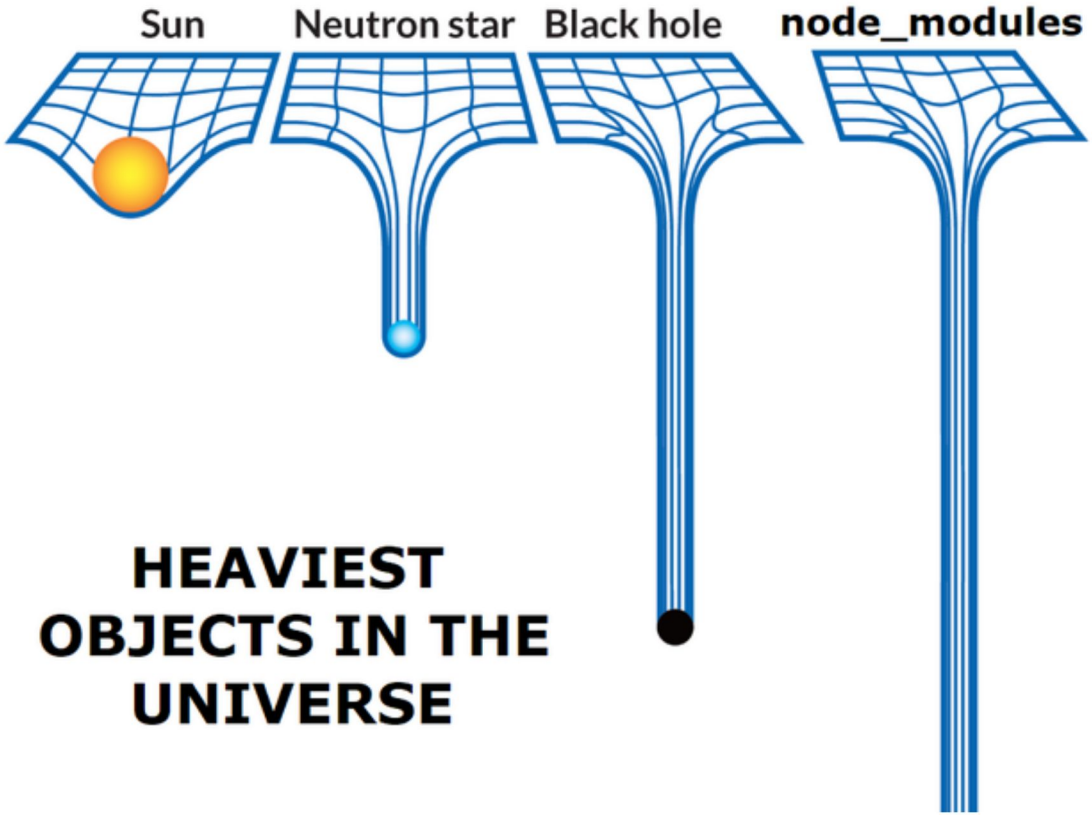
C Help

T

sometimes, this is someones first introduction to web dev!

problems with the web:

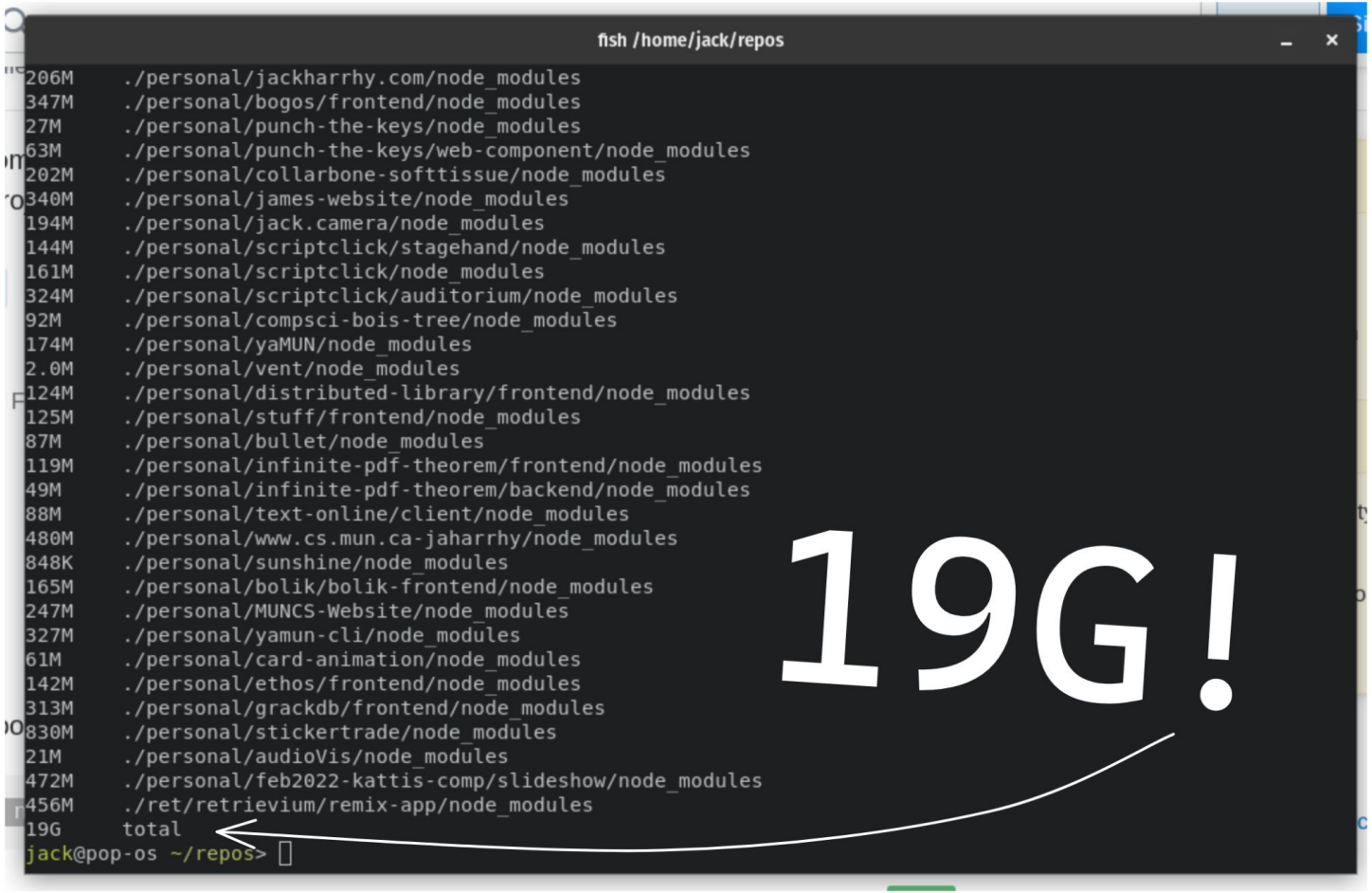
node_modules


Sun
Neutron star
Black hole
node_modules
HEAVIEST OBJECTS IN THE UNIVERSE

How to calculate size of all node_modules?

```
find . -type d -name node_modules -prune | tr '\n' '\0' |  xargs -0 du -sch
```

thanks stackoverflow


fish /home/jack/repos
206M ./personal/jackharrhy.com/node_modules
347M ./personal/bogos/frontend/node_modules
27M ./personal/punch-the-keys/node_modules
63M ./personal/punch-the-keys/web-component/node_modules
202M ./personal/collarbone-softtissue/node_modules
340M ./personal/james-website/node_modules
194M ./personal/jack.camera/node_modules
144M ./personal/scriptclick/stagehand/node_modules
161M ./personal/scriptclick/node_modules
324M ./personal/scriptclick/auditorium/node_modules
92M ./personal/compsci-bois-tree/node_modules
174M ./personal/yaMUN/node_modules
2.0M ./personal/vent/node_modules
124M ./personal/distributed-library/frontend/node_modules
125M ./personal/stuff/frontend/node_modules
87M ./personal/bullet/node_modules
119M ./personal/infinite-pdf-theorem/frontend/node_modules
49M ./personal/infinite-pdf-theorem/backend/node_modules
88M ./personal/text-online/client/node_modules
480M ./personal/www.cs.mun.ca-jaharrhy/node_modules
848K ./personal/sunshine/node_modules
165M ./personal/bolik/bolik-frontend/node_modules
247M ./personal/MUNCS-Website/node_modules
327M ./personal/yamun-cli/node_modules
61M ./personal/card-animation/node_modules
142M ./personal/ethos/frontend/node_modules
313M ./personal/grackdb/frontend/node_modules
830M ./personal/stickertrade/node_modules
21M ./personal/audioVis/node_modules
472M ./personal/feb2022-kattis-comp/slideshow/node_modules
456M ./ret/retrievium/remix-app/node_modules
19G total
jack@pop-os ~/repos>
19G!

problems with the web:

dependencies


leftpad

A man in Oakland, California, disrupted web development around the world last week by deleting 11 lines of code.

```
module.exports = leftpad;
function leftpad (str, len, ch) {
  str = String(str);
  var i = -1;
  if (!ch && ch !== 0) ch = ' ';
  len = len - str.length;
  while (++i < len) {
    str = ch + str;
  }
  return str;
}
```

Azer Koçulu

Image: GitHub

qz.com/646467/how-one-programmer-broke-the-internet-by-deleting-a-tiny-piece-of-code

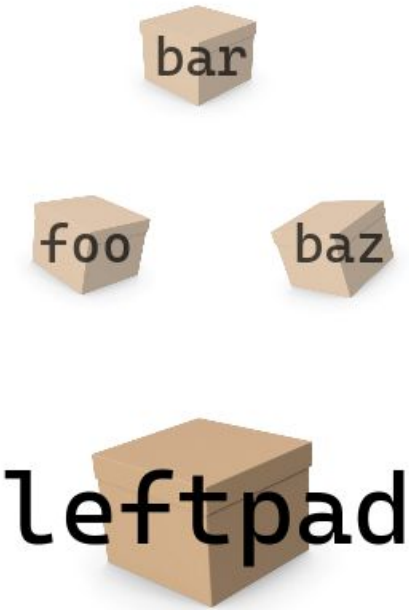
Azer had many packages on the npm registry, including one called "kik", which was unrelated to the actual messaging app by the same name

kik reached out to Azer to take over the ability to publish to the kik package, he declined

kik then reached out to npm directly, and they agreed to give up the package to them

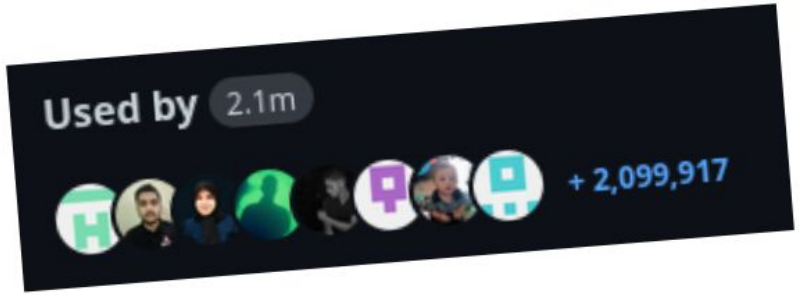
Azer Koçulu

Image: GitHub



npmjs.org tells me that left-pad is not available (404 page) #4

Closed **silkentrance** opened this issue on Mar 22, 2016 · 193 comments



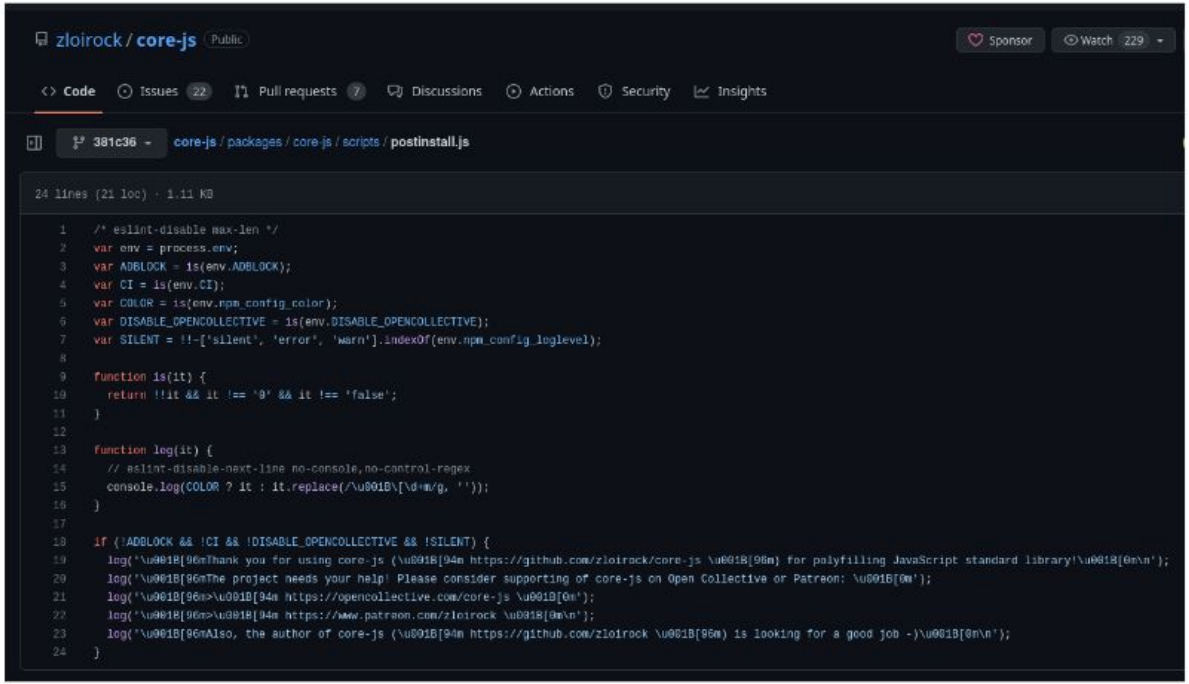
core-js

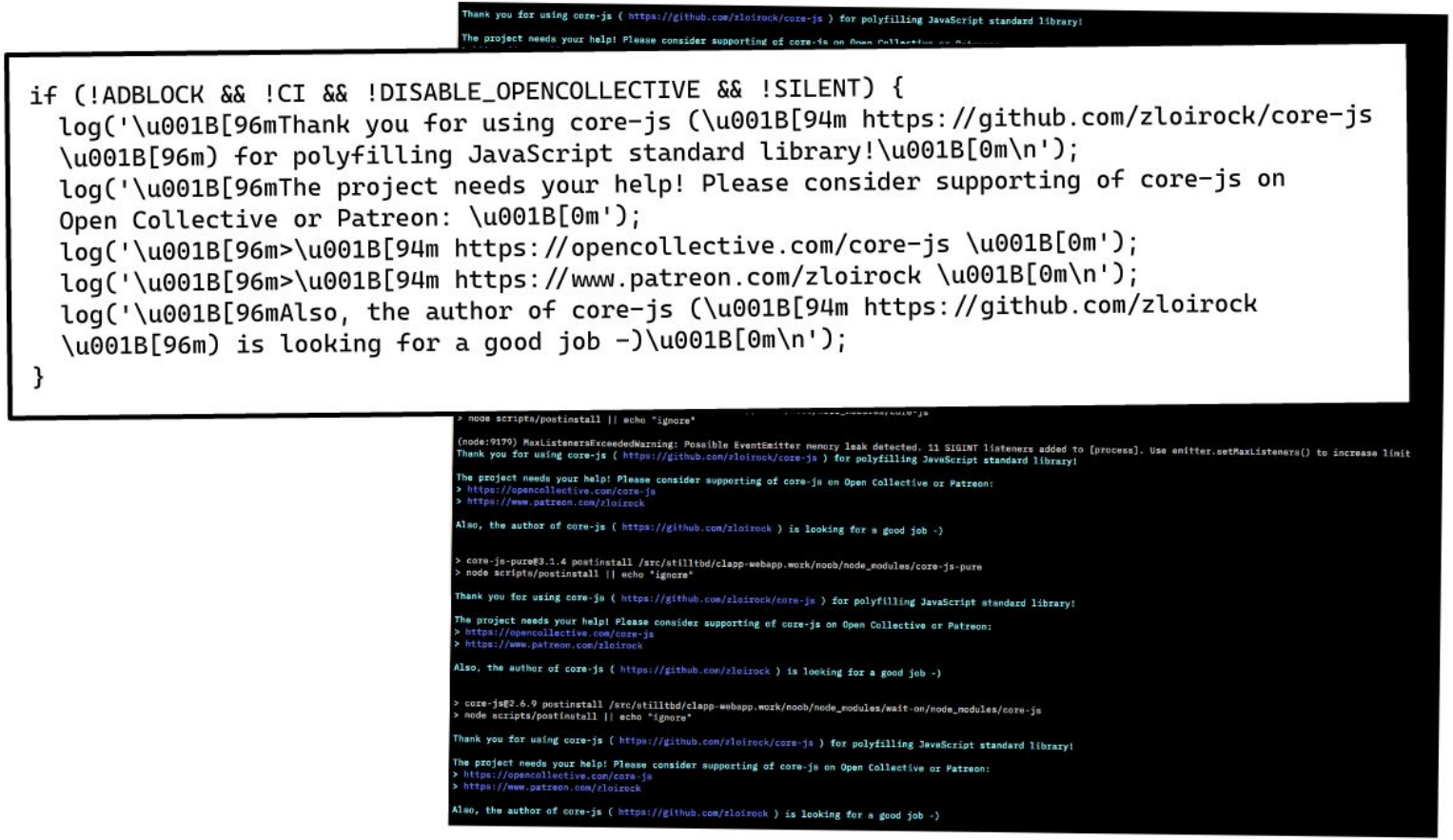
core-js

Modular standard library for JavaScript. Includes polyfills for ECMAScript up to 2023: promises, symbols, collections, iterators, typed arrays, many other features, ECMAScript proposals, some cross-platform WHATWG / W3C features and proposals like URL. You can load only required features or use it without global namespace pollution.

tl;dr: useful thing quite a few packages depend on

in june of 2019, the author of core-js added this to the 'post install' script within the package:





```
if (!ADBLOCK && !CI && !DISABLE_OPENCOLLECTIVE && !SILENT) {
  log('\u001B[96mThank you for using core-js (\u001B[94m https://github.com/zloirock/core-js
  \u001B[96m) for polyfilling JavaScript standard library!\u001B[0m\n');
  log('\u001B[96mThe project needs your help! Please consider supporting of core-js on
  Open Collective or Patreon: \u001B[0m');
  log('\u001B[96m>\u001B[94m https://opencollective.com/core-js \u001B[0m');
  log('\u001B[96m>\u001B[94m https://www.patreon.com/zloirock \u001B[0m\n');
  log('\u001B[96mAlso, the author of core-js (\u001B[94m https://github.com/zloirock
  \u001B[96m) is looking for a good job -)\u001B[0m\n');
}
```

```
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)


> core-js-pure@3.1.4 postinstall /src/stilltbd/clapp-webapp.work/noob/node_modules/core-js-pure
> node scripts/postinstall || echo "ignore"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)


> core-js@2.6.9 postinstall /src/stilltbd/clapp-webapp.work/noob/node_modules/wait-on/node_modules/core-js
> node scripts/postinstall || echo "ignore"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)
```

Get rid of postinstall message #548

Closed **jpike88** opened this issue on May 19, 2019 · 166 comments

jpike88 commented on May 19, 2019 • edited

My installation log isn't for advertising space. If you want donations, please do the right thing and promote yourself in places where appropriate. My logs are long enough as it is without this.

The right thing to do is to remove it.

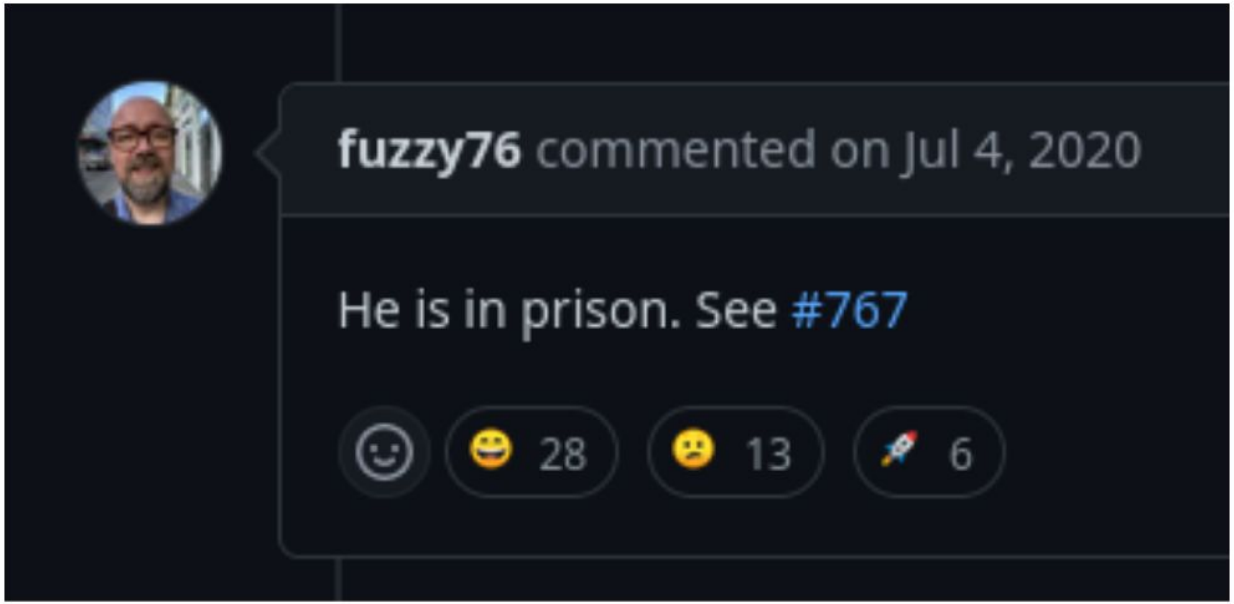
EDIT: An unmaintained fork is available: `core-js-without-ads`

👍 201 👎 635 😄 6 😕 41 👀 12

the notice was removed
from the repository

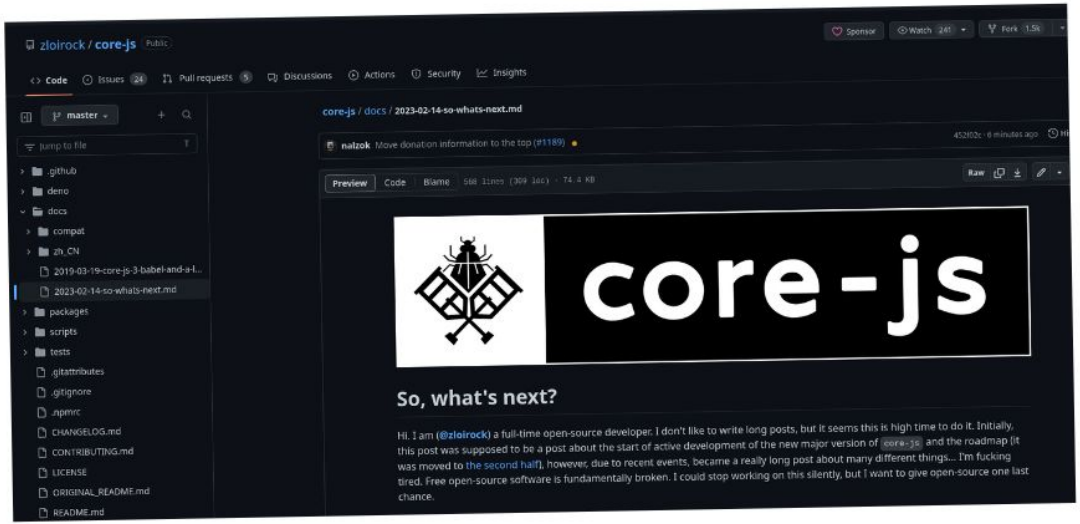
and some time passed ...


Did the author of core-js ever find a good job? #708
Closed
jannik-mohemian opened this issue on Nov 28, 2019 · 16 comments
jannik-mohemian commented on Nov 28, 2019
👋 I'm interested in whether you ever found a good job through the post-install message - it's been a while since this all started...
103
84
23
18
20


fuzzy76 commented on Jul 4, 2020
He is in prison. See #767
28
13
6

as of yesterday, the author of core-js posted a lengthy update called "So, what's next?"

https://github.com/zloirock/core-js/blob/master/docs/2023-02-14-so-whats-next.md

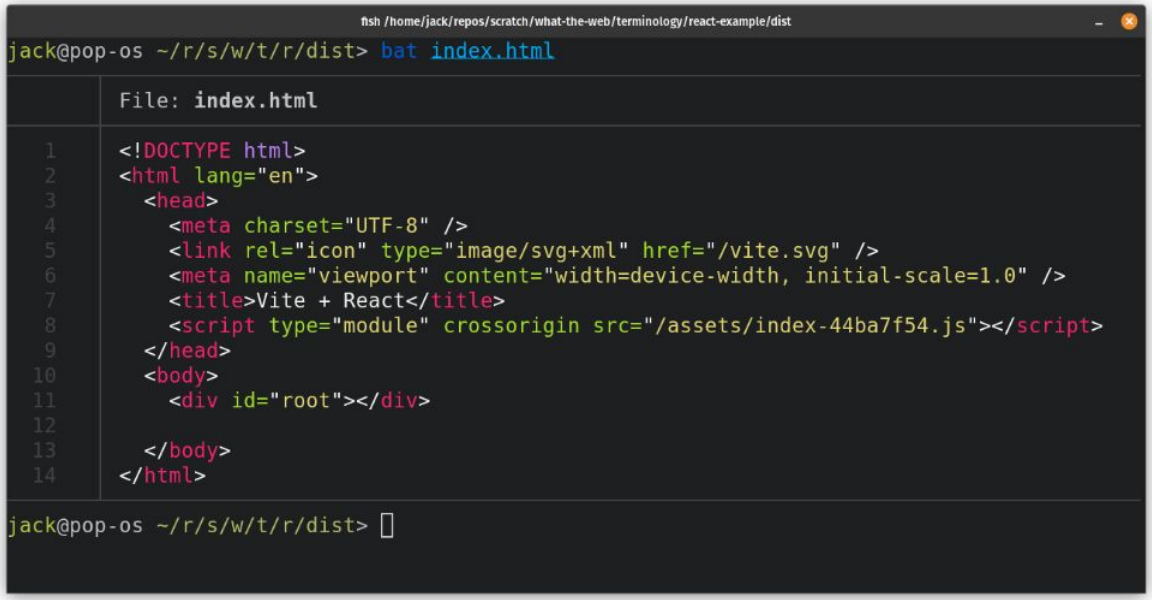


sadly, these two stories
are hardly isolated incidents

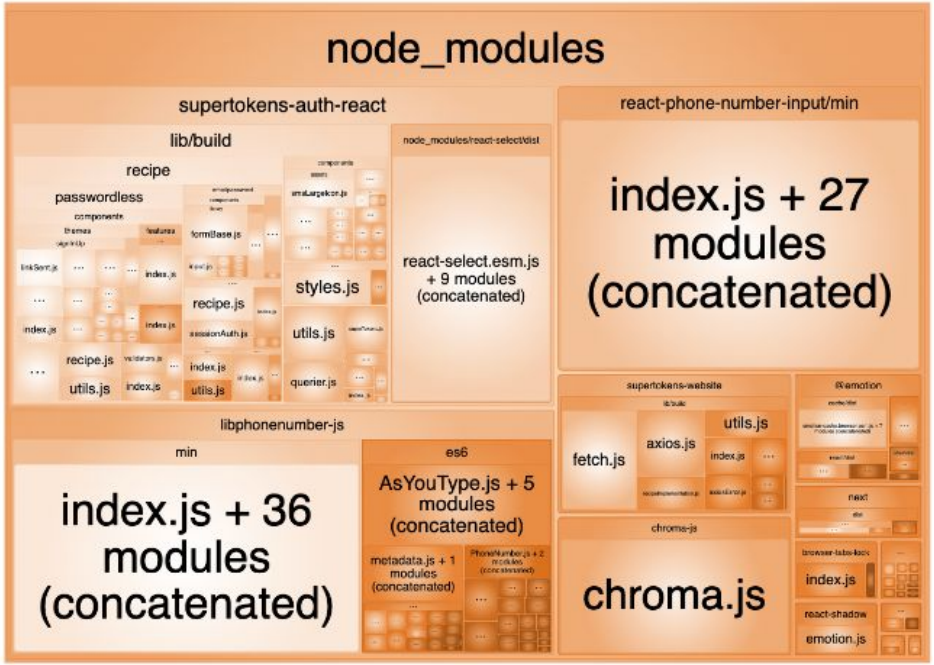
npm ecosystem

problems with the web:

single page applications


fish /home/jack/repos/scratch/what-the-web/terminology/react-example/dist
jack@pop-os ~/r/s/w/t/r/dist> bat index.html
File: index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <title>Vite + React</title>
8 <script type="module" crossorigin src="/assets/index-44ba7f54.js"></script>
9 </head>
10 <body>
11
12
13 </body>
14 </html>
jack@pop-os ~/r/s/w/t/r/dist>

large client bundles!



52	60	73	86	PWA
Performance	Accessibility	Best Practices	SEO	Progressive Web App

52

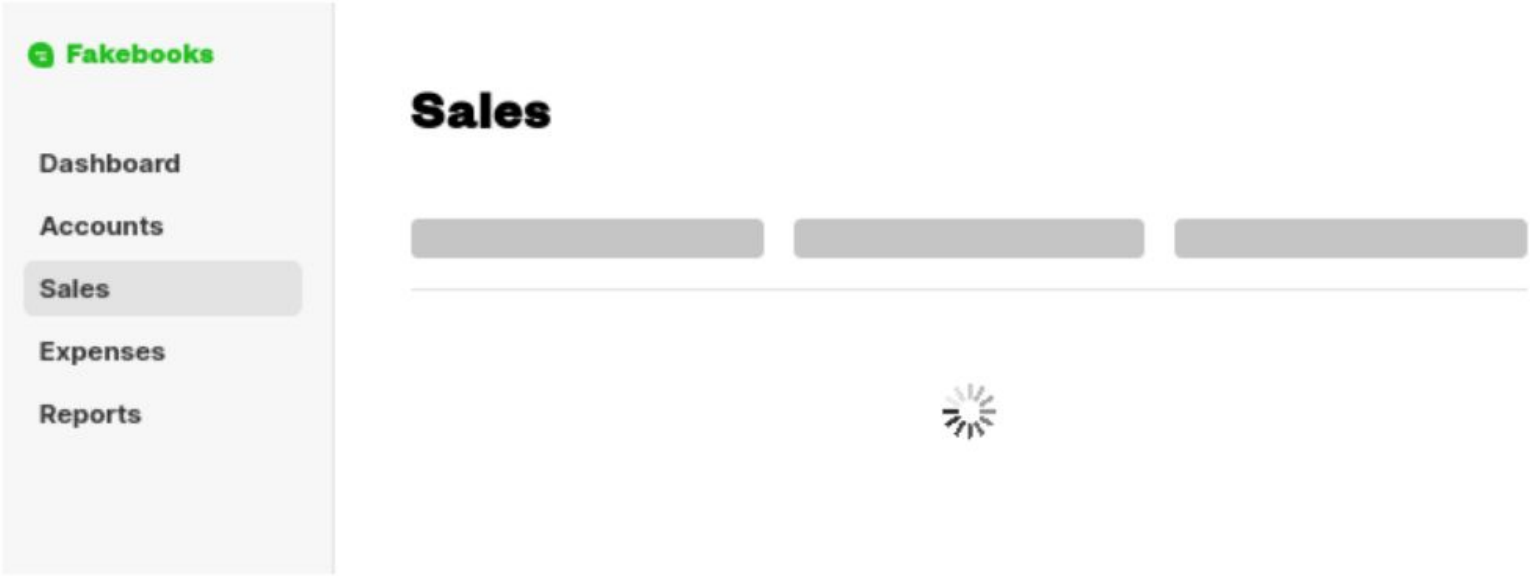
Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

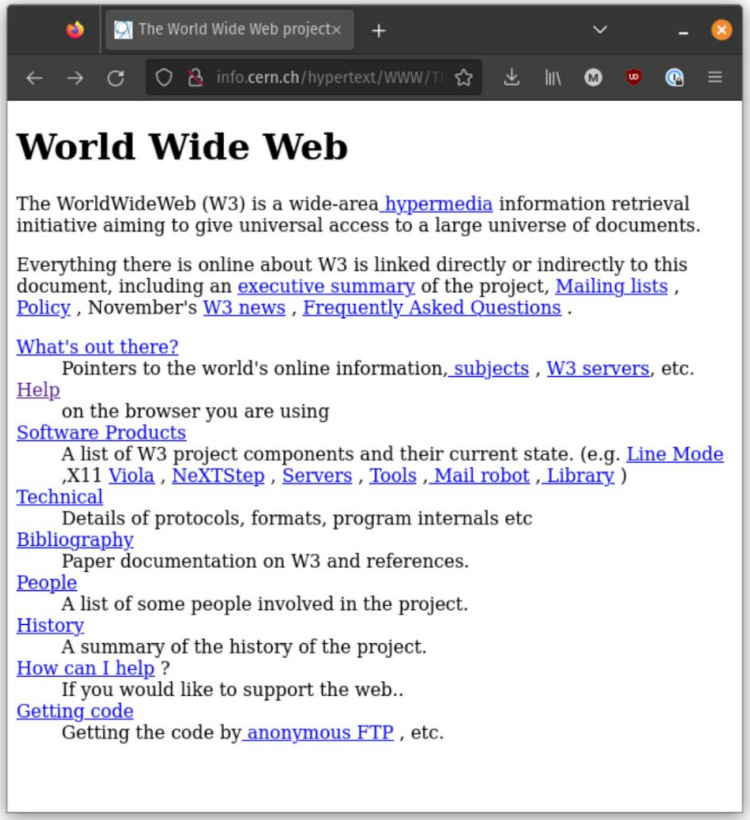
problems with the web:

loading spinners

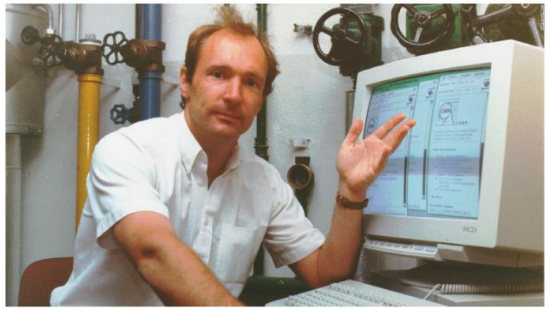


i'm looking at you, ClickUp ...

how did we get here?



Tim Berners-Lee



HTML draft version timeline

October 1991

HTML Tags,[6] an informal CERN document listing 18 HTML tag

June 1992

First informal draft of the HTML DTD,[32] with seven[33][34][35] s August 18, Nov November 20, November 22)

November 1992

HTML DTD 1.1 (the first with a version number, based on RCS r 1.0), an info

June 1993

Hypertext Markup Language[36] was published by the IETF IIIR V ugh proposal replaced by a second version[37] one month later.

November 1993

HTML+ was published by the IETF as an Internet Draft and wa : Markup La 1994.[38]

November 1994

First draft (revision 00) of HTML 2.0 published by IETF itself[39] (c), that fina RFC 1866 in November 1995.[41]

April 1995 (authored March 1995)

HTML 3.0[42] was proposed as a standard to the IETF, but the pro ember 1 included many of th apabilities that were in Raggett's HTML+ p flow ar complex mathema rmulas.[43]

W3C began deve its own Arena browser as a test bed f HTML 3 and Cascading Style Sheets,[44][45][46] but HTML 3.0 for several reas was considered very large at 150 pages and the pace of browser development, as well as the number of interested parties, had o urces of the IETF.[13] Browser vendors, including Microsoft and Netscape at the time, chose to implement different subsets of HT 's draft fea as well as to introduce their own extensions to it.[13] (see Browser wars). These included extensions to control stylistic asp docume ry to the "belief [of the academic engineering community] that such things as text color, background texture, fo fon finitely outside the scope of a language when their only intent was to specify how a document would be organize n a W3C Fellow for many years, has commented for example: "To a certain extent, Microsoft built its busine Web by extending HTML features."[13]

JavaScript

Screenshot of JavaScript source code with HTML

Paradigm	Multi-paradigm: event-driven, functional, imperative, procedural, object-oriented programming
Designed by	Brendan Eich of Netscape initially; others have also contributed to the ECMAScript standard
First appeared	December 4, 1995; 27 years ago[1]

Cascading Style Sheets (CSS)

The official logo of the latest version, CSS 3

Filename extension	`.css`
Internet media type	`text/css`
Uniform Type Identifier (UTI)	public.css
Developed by	World Wide Web Consortium (W3C)
Initial release	17 December 1996; 25 years ago

Server-side rendering!

PHP: PHP Hypertext Preprocessor

PHP: Personal Home Page

First appeared: June 8, 1995; 27 years ago

One of the most significant breakthroughs for the web was going from this:

```
<html>
    <body>
        This document has been prepared ahead of time.
        Regards.
    </body>
</html>
```

To having easily embedded dynamic values:

```
<html>
    <body>
        Y2K? <?php echo time(); ?>
    </body>
```

Java applet

From Wikipedia, the free encyclopedia
(Redirected from Java applets)

Java applets were small applica hat
programming language, or anot f Java
compiles to Java bytecode, and deliv and
bytecode. The user launched the Jav M) in
the applet was then executed within could
a process separate from the web b , Sun's
appear in a frame of the web page
AppletViewer, or a stand-alone to

ActiveX

activex

Original author(s)	Microsoft
Developer(s)	Microsoft
Initial release	1996; 26 years ago

Adobe Flash

Developer(s)	Harman (2021–present for enterprise users) [1][2][3][4] Zhongcheng (2017–present in China)[5] Adobe Inc. (2005–2020) Macromedia (1996–2005) FutureWave (1993–1996)
Target	Web browsers. iOS (via

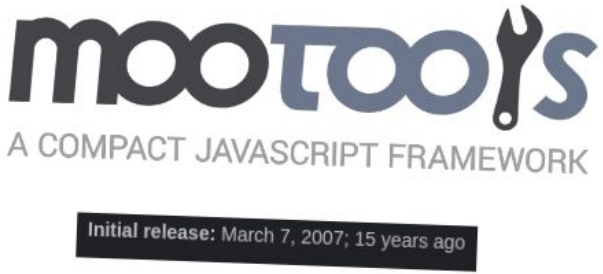
Initial release date: August 2004

Initial release date: July 21, 2005

Initial release date: May 27, 2009

the client side got busier over time, but browsers were frustrating when it came to shared consistent functionality

so, tools like jQuery and MooTools were created to help bridge the gap between browsers and give useful utilities for developers




ANGULARJS
by Google


ember


BACKBONE.JS


METEOR


Gulp


GRUNT

World Wide Web Consortium (W3C)

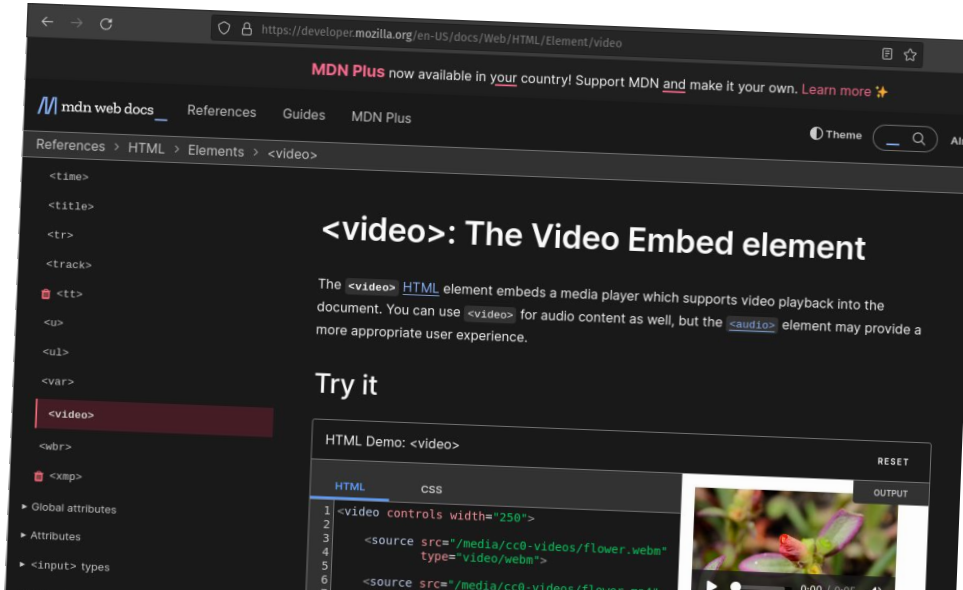
Founder: Tim Berners-Lee

Founded: October 1, 1994

Web Hypertext Application Technology Working Group (WHATWG)

Founded: June 4, 2004


HTML


https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video
MDN Plus now available in your country! Support MDN and make it your own. Learn more
mdn web docs
References
Guides
MDN Plus
Theme
References › HTML › Elements › <video>
<time>
<title>
<tr>
<track>
<tt>
<u>

<var>
<video>
<wbr>
<xmp>
Global attributes
Attributes
<input> types
<video>: The Video Embed element
The <video> HTML element embeds a media player which supports video playback into the document. You can use <video> for audio content as well, but the <audio> element may provide a more appropriate user experience.
Try it
HTML Demo: <video>
RESET
HTML
CSS
OUTPUT
<video controls width="250">
<source src="/media/cc0-videos/flower.webm"
type="video/webm">
<source src="/media/cc0-videos/


Java applet
From Wikipedia, the free encyclopedia
(Redirected from Java applets)
Java applets were small
programming language, or
compiles to Java bytecode,
bytecode. The user launched
the applet was then executed
a process separate from the
appear in a frame of the
AppletViewer, or a stand-alone
JAVA
ActiveX
activex
Original author(s) Microsoft
Developer(s) Microsoft
Initial release 1996; 26 years ago
Adobe Flash
Developer(s) Harman (2021–present for enterprise users)
Zhongcheng (2017–present in China)
Adobe Inc. (2005–2020)
Macromedia (1996–2005)
FutureWave (1993–1996)
Target Web browsers, iOS (via

thing aren't that bad

thing aren't that bad:

we (seem) to have gotten better about dependencies

Unpublishing packages from the registry

As a package owner or collaborator, if your package has no dependents, you can permanently remove it from the npm registry by using the CLI. You can unpublish within 72 hours of the initial publish; beyond 72 hours, you can still unpublish your package if it meets certain criteria.

These criteria are set to avoid damaging the JavaScript package ecosystem. If you cannot unpublish your package, you can deprecate it instead.

How the maintainers of urllib3 keep the project secure and healthy →

For maintainers | Login

TIDELIFT

Maximize the health and security of the open source powering your applications

A proactive approach to managing open source

Tidelift provides the tools, data, and strategies that help organizations assess risk and improve the health, security, and resilience of the open source used in their applications.

Backed maintai

Tidelift partne
ensure the op
enterprise sta

npm Docs

Search npm Docs

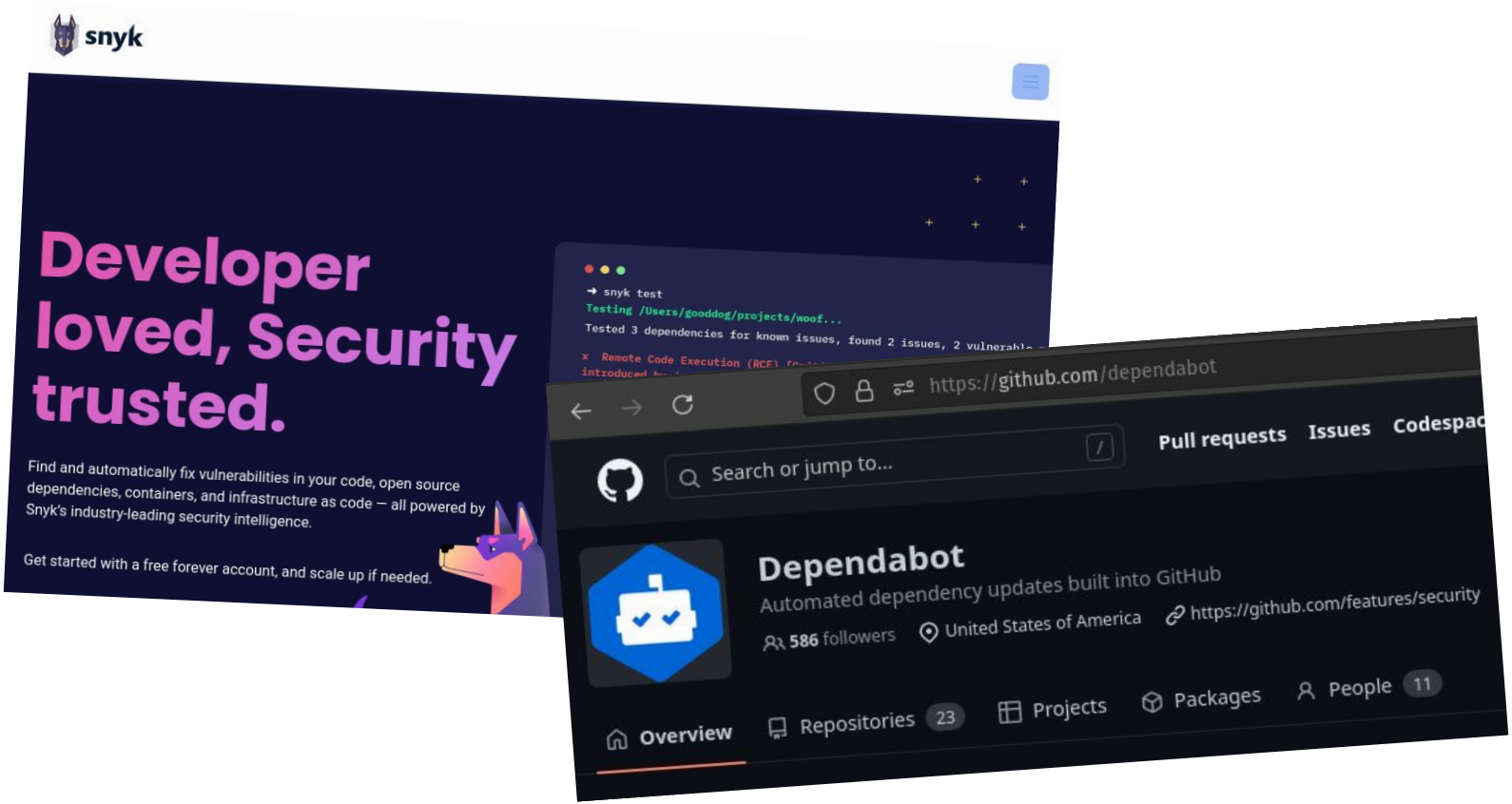
- About npm
- Getting started
- Packages and modules
- Integrations

npm-fund

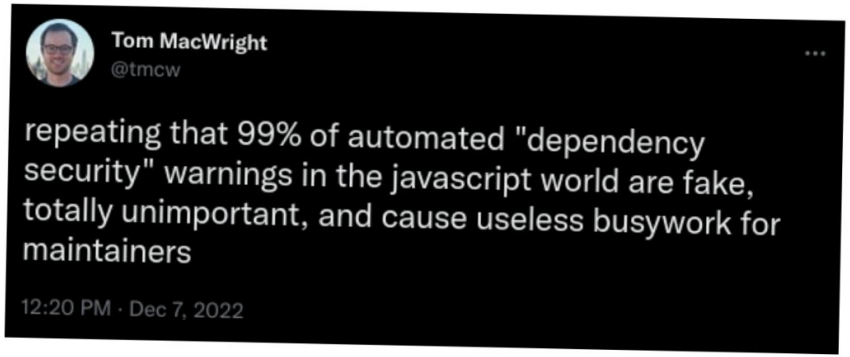
Retrieve funding information

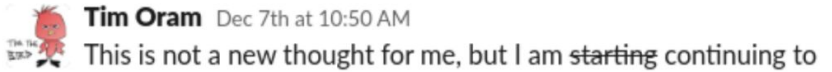
Select CLI Version:

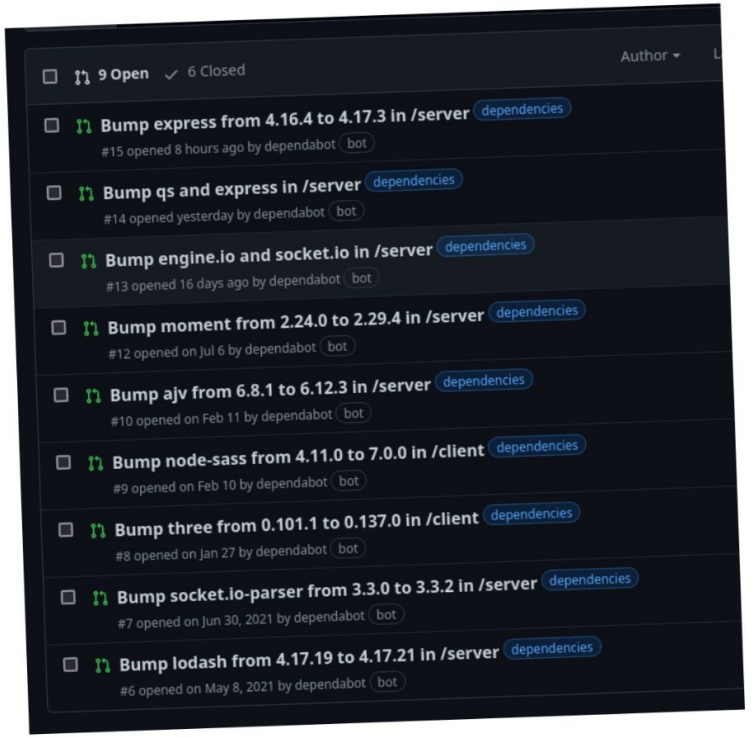
Version 7.24.2 (Legacy Release)


snyk
Developer loved, Security trusted.
Find and automatically fix vulnerabilities in your code, open source dependencies, containers, and infrastructure as code — all powered by Snyk's industry-leading security intelligence.
Get started with a free forever account, and scale up if needed.
snyk test
Testing /Users/gooddog/projects/woof...
Tested 3 dependencies for known issues, found 2 issues, 2 vulnerable
Remote Code Execution (RCE)
https://github.com/dependabot
Search or jump to...
Pull requests
Issues
Dependabot
Automated dependency updates built into GitHub
586 followers
United States of America
https://github.com/features/security
Overview
Repositories 23
Projects
Packages
People 11

but, sometimes the signal to noise ratio isn't that great ...

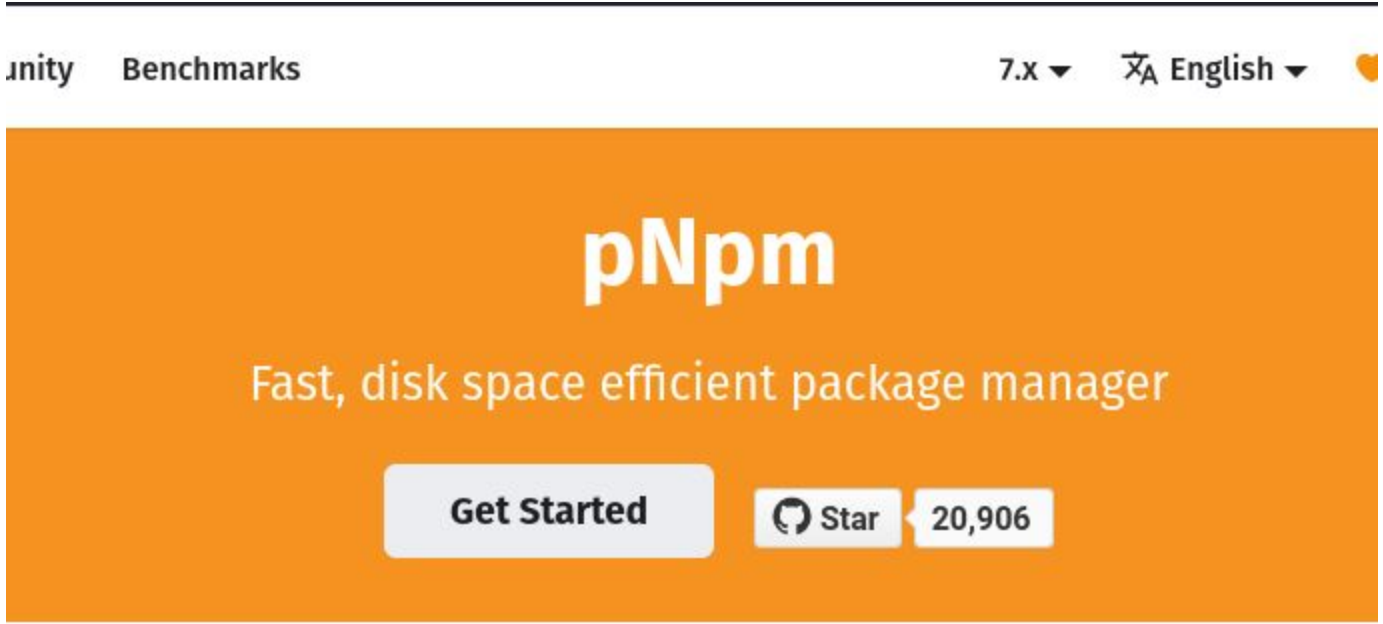






thing aren't that bad:

pnpm helps solve the many node_modules issue

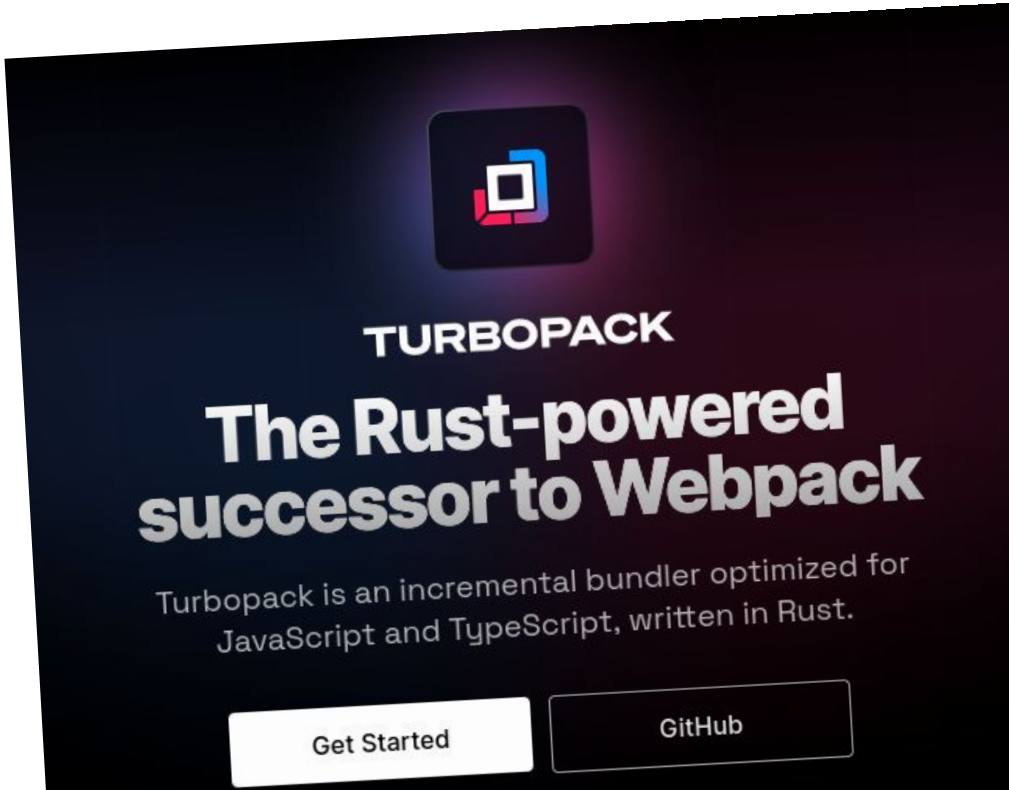

unity
Benchmarks
7.x
English
pNpm
Fast, disk space efficient package manager
Get Started
Star
20,906

thing aren't that bad:

the build system story is a lot better than it used to be


Create React App
Set up a modern web app by running one command.
Get Started

```
fish /home/jack/repos/scratch/what-the-web/terminology
jack@pop-os ~/r/s/w/terminology> npm create vite@latest
✔ Project name: … react-example
✔ Select a framework: › React
✔ Select a variant: › JavaScript

Scaffolding project in /home/jack/repos/scratch/what-the-web/terminology/react-example...

Done. Now run:

  cd react-example
  npm install
  npm run dev

jack@pop-os ~/r/s/w/terminology>
```


TURBOPACK
The Rust-powered successor to Webpack
Turbopack is an incremental bundler optimized for JavaScript and TypeScript, written in Rust.
Get Started
GitHub

thing aren't that bad:

you can still have a fun dev experience without shipping any js


HUGO


jekyll


Gatsby


NuxtJS


astro

thing aren't that bad:

the industry seems less focused on single page apps


NEXT.JS


Remix

tip: try out Astro

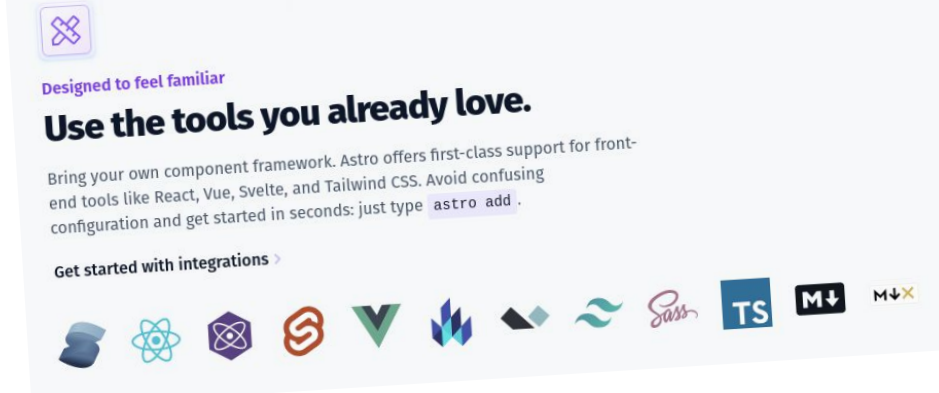
of all of the static site generators out there right now, Astro has to be my favorite, both because of how simple it is out of the box, but also all the features it packs in too

src/components/DynamicHtml.astro

```astro
---
const items = ["Dog", "Cat", "Platypus"];
---
<ul>
  {items.map((item) => (
    <li>{item}</li>
  ))}
</ul>
```

src/components/MyComponent.astro

```astro
---
import SomeAstroComponent from '../components/SomeAstroComponent.astro';
import SomeReactComponent from '../components/SomeReactComponent.jsx';
import someData from '../data/pokemon.json';

// Access passed-in component props, like `<X title="Hello, World" />`
const {title} = Astro.props;
// Fetch external data, even from a private API or database
const data = await fetch('SOME_SECRET_API_URL/users').then(r => r.json());
---
<!-- Your template here! -->
```



```
---
import Mug from "../components/Mug.astro";
import MainLayout from "../layouts/MainLayout.astro";
---

<MainLayout title="about">
    <div class="mb-4">
        <Mug />
    </div>
    <ul>
        <h1>hi, i'm jack</h1>
        <li class="pt-3">
            currently i'm working at <a href="http
            >, working on super neat web-based geo
        </li>
        <li class="pt-3">
            previously i've worked for
            <a href="https://bluedropism.com/">Blue
                href="https://www.colabsoftware.com
            >, as a full stack developer
        </li>
        <li class="pt-3">
            instructor at <a href="https://www.get-c
        </li>
        <li class="text-sm pt-3">
            <span class="italic opacity-50">(planning
            undergrad at <a href="https://www.mun.ca/
            <span class="italic opacity-50">(eventuall
        </li>
        <li class="text-sm pt-3">
            active with <a href="https://ctsnl.ca/">CTS
            <a href="https://www.meetup.com/NDevMeetup/
        </li>
        <li class="text-sm pt-3">
            building things with pals in
            <a href="https://github.com/fogo-sh">fogo.sh
        </li>
        <li class="text-sm pt-3">from wales 🏴, vibing in
```

about - jackharrhy

https://jackharrhy.dev

github

twitter

instagram

steam

youtube

hi, i'm jack

currently i'm working at C-CORE, working on super neat web-based geospatial sofware

previously i've worked for Bluedrop, and CoLab Software, as a full stack developer

instructor at Get Coding

(planning on) finishing up my undergrad at MUN *(eventually)*

active with CTSNL, and sometimes NDev

building things with pals in fogo.sh

from wales 🏴, vibing in canada 🇨🇦

my resume, built as a modular latex resume

contact me:

me@jackharrhy.com

jack arthur null#7539

tip: use Vite

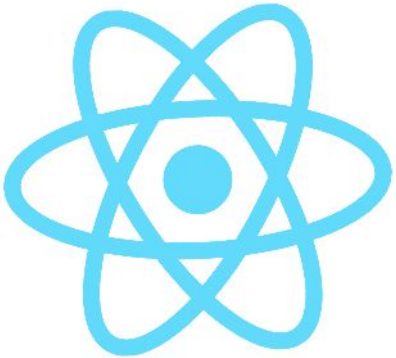
if you are planning on starting a new frontend project, or have a existing frontend project you want to speed up / move away from create-react-app/webpack, Vite is the way to go

tip: use Vitest

in the same ecosystem as Vite, is Vitest, a test runner that comes out of the box with support for most things you could think of without configuration
Vitest is jest compatible, so if you're already using that, you're set

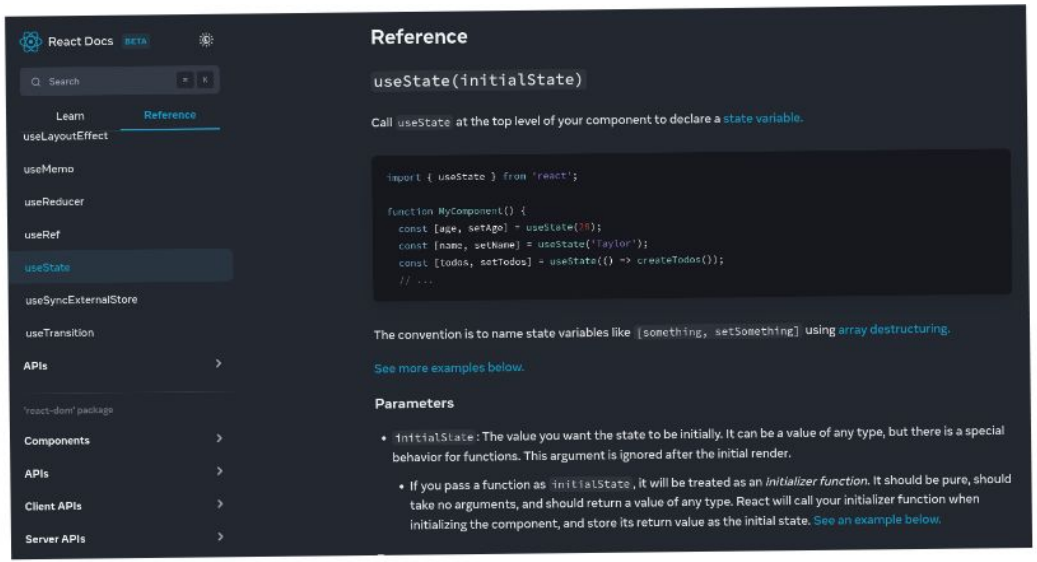
React is still a safe bet as framework of choice

while there are other choices out there, and some newer frameworks with some pretty cool demos, as far as what I and most would recommend picking as a framework for some time, and now, is React



tip: use the new beta.reactjs.org site

the old docs still reference class components everywhere, the beta docs (soon hopefully to replace the official docs) are much more improved and modern in comparison



tip: use TanStack Query for any & all data fetching

the application you have, likely has to do some async stuff to fetch data, which react doesn't give you a perfect handle for yet

(Suspense is on the way, but still not perfect, and hardly concerns itself with state to the degree TanStack Query does)

```jsx
// 1. Import *useState* and *useEffect*
import React, {useState, useEffect} from 'react';
import './App.css';

function App() {
    // 2. Create our *dogImage* variable as well as the *setDogImage* function via useState
    // We're setting the default value of dogImage to null, so that while we wait for the
    // fetch to complete, we dont attempt to render the image
  let [dogImage, setDogImage] = useState(null)

    // 3. Create out useEffect function
  useEffect(() => {
    fetch("https://dog.ceo/api/breeds/image/random")
    .then(response => response.json())
        // 4. Setting *dogImage* to the image url that we received from the response above
    .then(data => setDogImage(data.message))
  },[])

  return (
    <div className="App">
        {/* 5. Using *dogImage as* the *src* for our image*/}
    {dogImage && <img src={dogImage}></img>}
    </div>
  );
}

export default App;
```

a single fetch within a useEffect, is sadly not enough

```
function Todos() {
  const { isLoading, isError, data, error } = useQuery({
    queryKey: ['todos'],
    queryFn: fetchTodoList,
  })

  if (isLoading) {
    return <span>Loading...</span>
  }

  if (isError) {
    return <span>Error: {error.message}</span>
  }

  // We can assume by this point that `isSuccess === true`
  return (
    <ul>
      {data.map((todo) => (
        <li key={todo.id}>{todo.title}</li>
      ))}
    </ul>
  )
}
```

basic api, you can pass it a proper async function, and it handles quite a bit for you!

Out of the box, TanStack Query is configured with **aggressive but sane** defaults. **Sometimes these defaults can catch new users off guard or make learning/debugging difficult if they are unknown by the user.** Keep them in mind as you continue to learn and use TanStack Query:

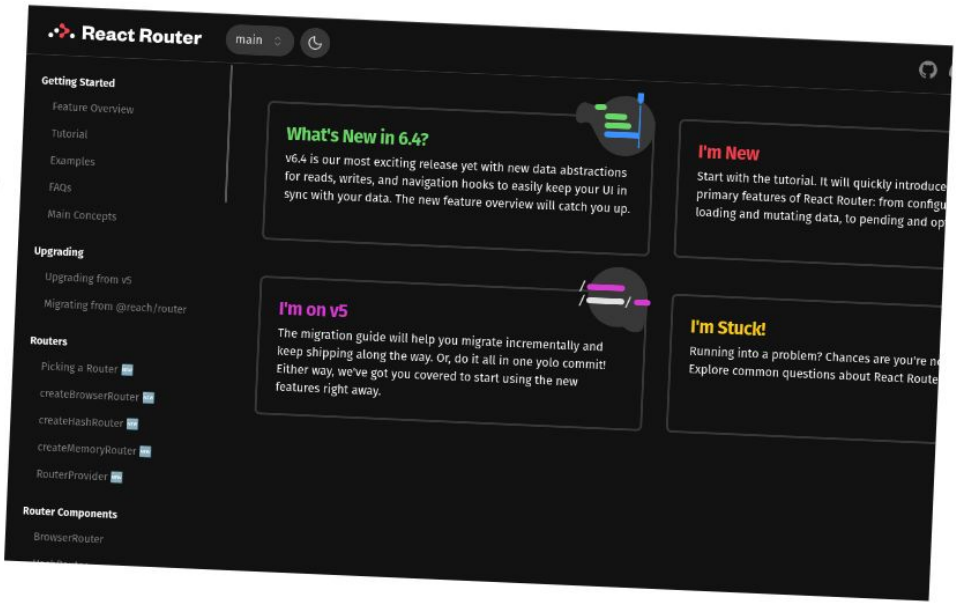
- Query instances via `useQuery` or `useInfiniteQuery` by default **consider cached data as stale**.

> ***"To change this behavior, you can configure your queries both globally and per-query using the*** `staleTime` ***option. Specifying a longer*** `staleTime` ***means queries will not refetch their data as often"***

- Stale queries are refetched automatically in the background when:
  - New instances of the query mount
  - The window is refocused
  - The network is reconnected
  - The query is optionally configured with a refetch interval

tip: use React Router,
especially the 6.4+ features

if you're writing a react app, chances are it already uses React Router to perform client side navigation, however there are some new features you can make use of that allows it to handle data loading/mutations :)



```jsx
import { useLoaderData } from "react-router-dom";
import { getContacts } from "../contacts";

export async function loader() {
  const contacts = await getContacts();
  return { contacts };
}

export default function Contacts() {
  const { contacts } = useLoaderData();
  return (
    <>
      <div id="sidebar">
        <h1>React Router Contacts</h1>
        {/* Code to render contacts */}
      </div>
    </>
  );
}
```

look ma, component
with no loading state!

```jsx
import { useLoaderData } from "react-router-dom";
import { getContacts } from "../contacts";

// ...
export async function action() {
  const contact = await createContact();
  return { contact };
}

export default function Contacts() {
  // ...
  return (
    <>
      <div id="sidebar">
        <h1>React Router Contacts</h1>
        {/* Code to render contacts */}
        <Form method="post">
          <button type="submit">New</button>
        </Form>
      </div>
    </>
  );
}
```

boring old forms, back again in style!

use TypeScript

you should have a reason
you are not using TypeScript,
and it should be a good one,
if not you should figure out
a pathway for you/your team to
be using TypeScript

Kent C. Dodds
@kentcdodds

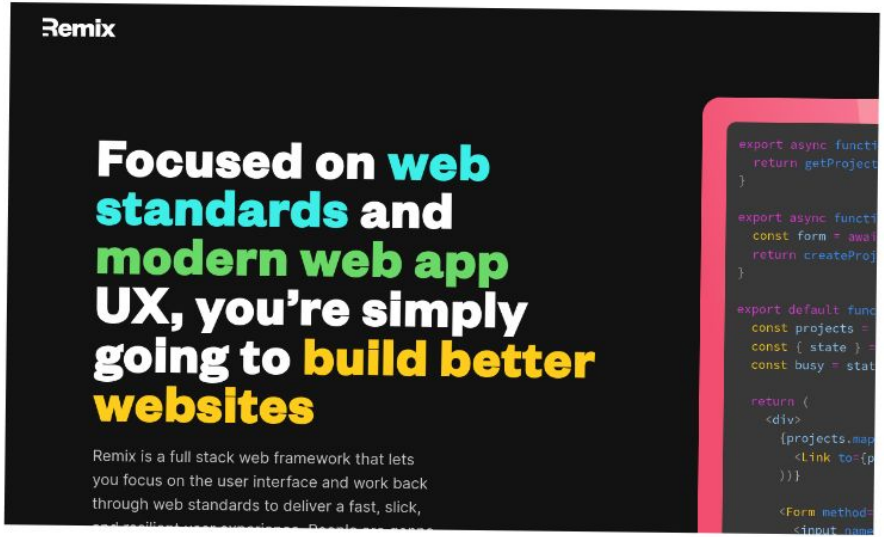
Unless I'm teaching you JavaScript fundamentals, everything I teach will be taught with TypeScript going forward.

TypeScript has won, and it's only a matter of time you're using it whether you like it or not.

10:53 PM · Feb 11, 2023 · **746.1K** Views

tip: use Remix

while some would consider Next.js the bigger player here, I myself am a big fan of the approach Remix to server side React

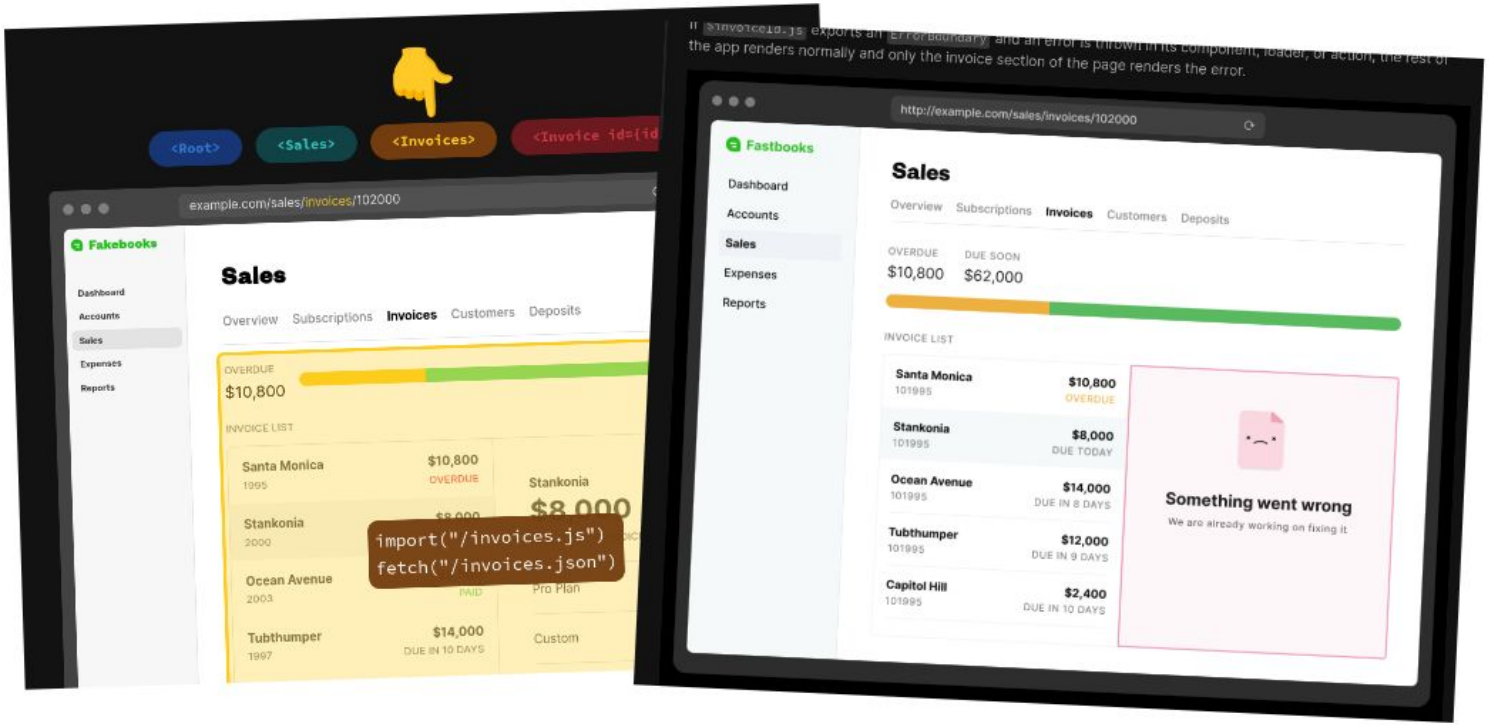


```tsx
export async function loader({ request }) {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({ title: form.get("title") });
}

export default function Projects() {
  const projects = useLoaderData();
  const { state } = useTransition();
  const busy = state === "submitting";

  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug}>{project.title}</Link>
      ))}

      <Form method="post">
        <input name="title" />
        <button type="submit" disabled={busy}>
          {busy ? "Creating..." : "Create New Project"}
        </button>
      </Form>
    </div>
  );
}
```

this looks quite similar to the previous react router example (mostly because its the same folk behind it),

HOWEVER, the loader and action functions you see here, are on the backend!

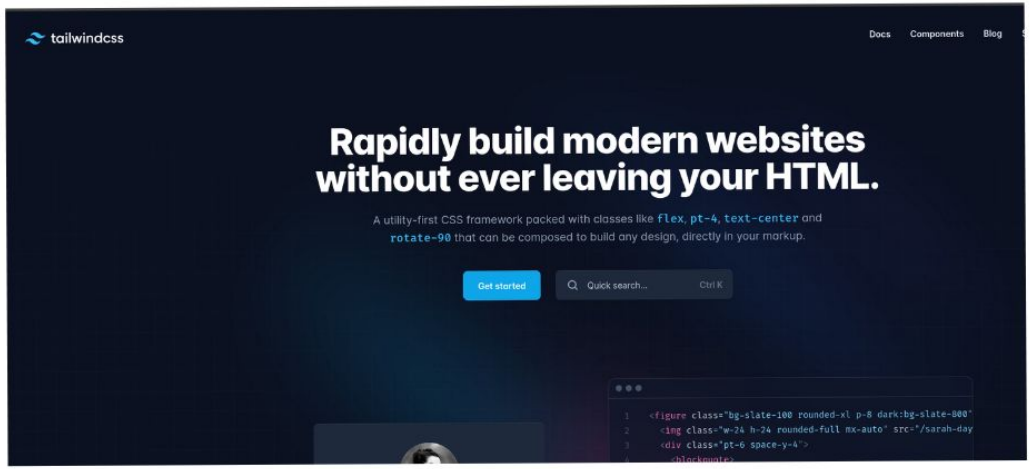
nested routing, nested error boundaries,
streaming, client only / server only code,
i could probably squeeze out a standalone Remix talk
in itself

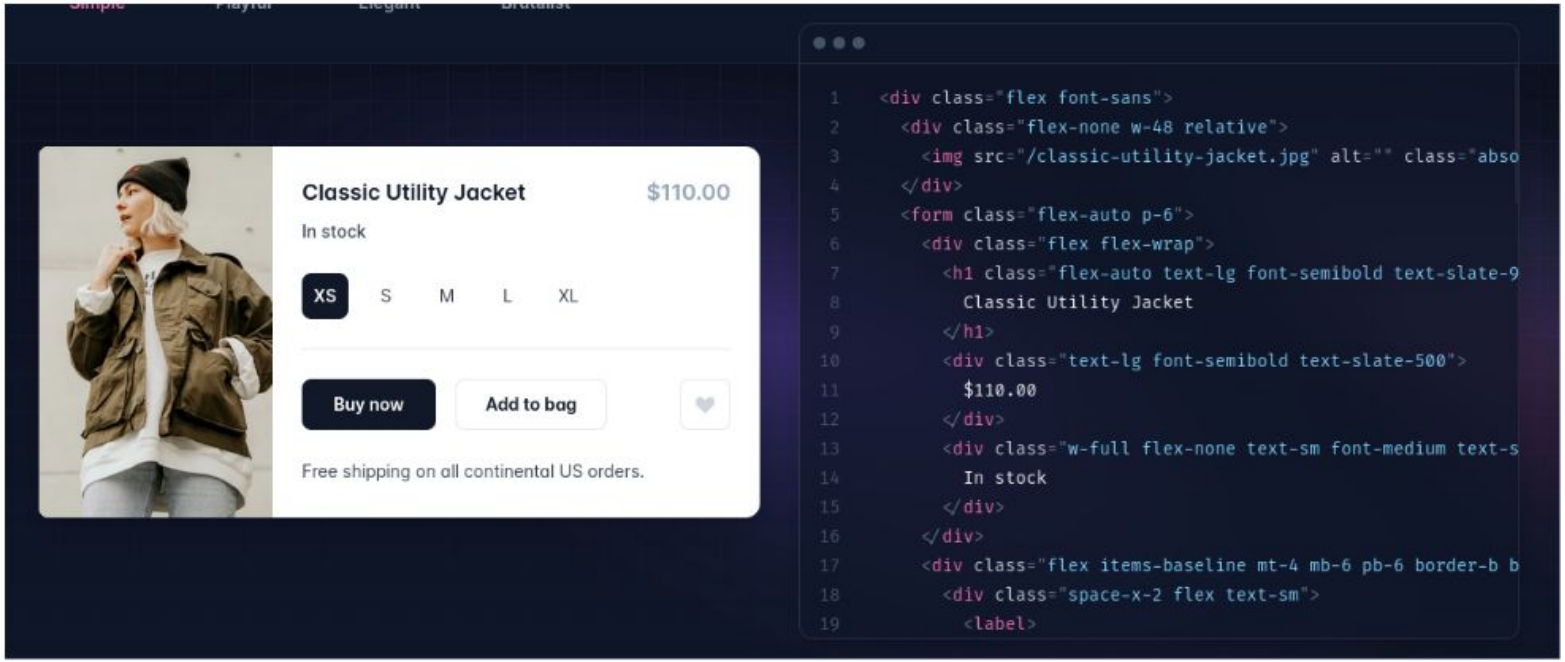


tip: give Tailwind CSS a try,
its cursed, and you might love it

i haven't talked about styling a lot this talk, but i might drop a mention to Tailwind CSS as an interesting new approach to styling elements / building out a design system

tailwindcss




Classic Utility Jacket
$110.00
In stock
XS
S
M
L
XL
Buy now
Add to bag
Free shipping on all continental US orders.

<img src="/classic-utility-jacket.jpg" alt="" class="abso

<form class="flex-auto p-6">

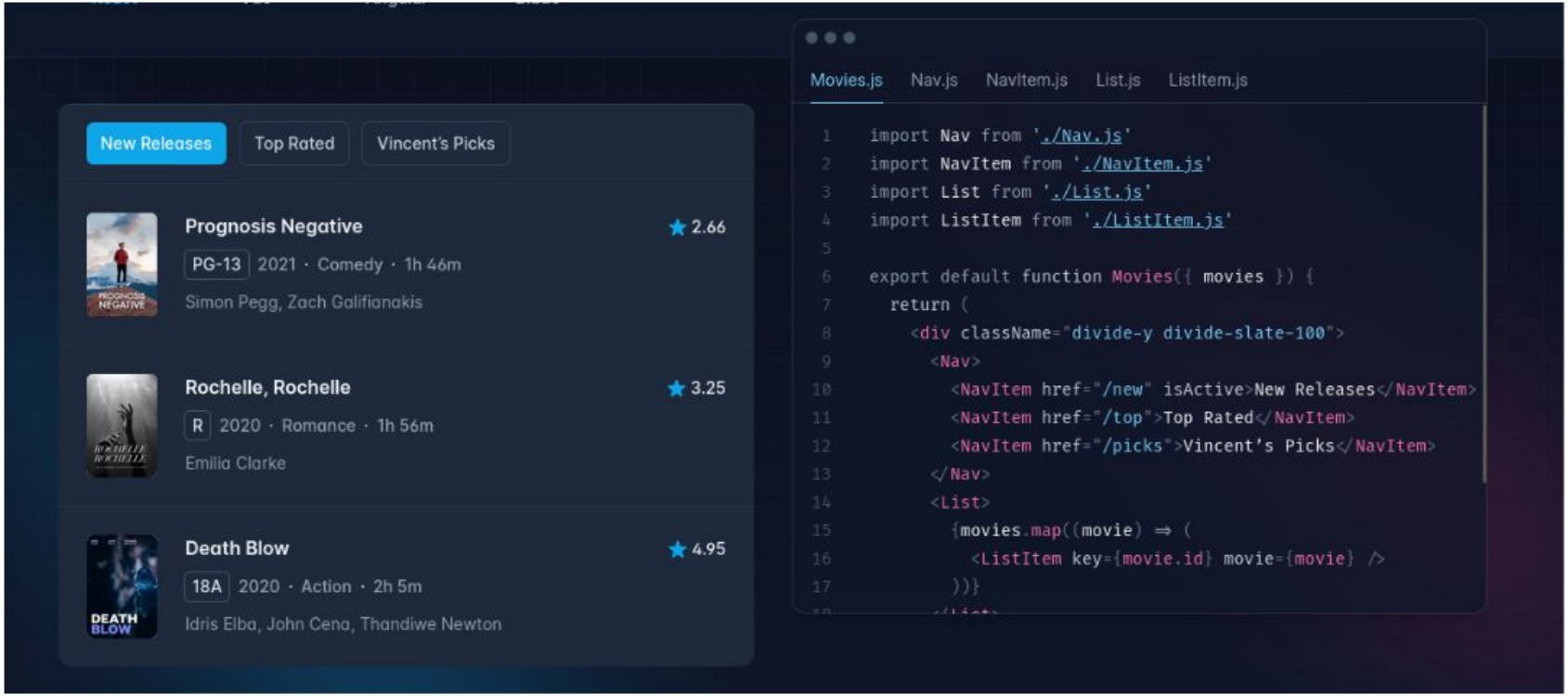
<h1 class="flex-auto text-lg font-semibold text-slate-9
Classic Utility Jacket
</h1>

$110.00

<div class="w-full flex-none text-sm font-medium text-s
In stock

<div class="flex items-baseline mt-4 mb-6 pb-6 border-b b

<label>


New Releases
Top Rated
Vincent's Picks
Prognosis Negative
2.66
PG-13 2021 · Comedy · 1h 46m
Simon Pegg, Zach Galifianakis
Rochelle, Rochelle
3.25
R 2020 · Romance · 1h 56m
Emilia Clarke
Death Blow
4.95
18A 2020 · Action · 2h 5m
Idris Elba, John Cena, Thandiwe Newton
Movies.js
Nav.js
NavItem.js
List.js
ListItem.js
import Nav from './Nav.js'
import NavItem from './NavItem.js'
import List from './List.js'
import ListItem from './ListItem.js'
export default function Movies({ movies }) {
return (

<Nav>
<NavItem href="/new" isActive>New Releases</NavItem>
<NavItem href="/top">Top Rated</NavItem>
<NavItem href="/picks">Vincent's Picks</NavItem>
</Nav>
<List>
{movies.map((movie) => (
<ListItem key={movie.id} movie={movie} />
))}

tip: use vim keybindings btw

I'm Jack Harrhy

And this was

why the web

questions/comments/critiques?

interested in chatting?
feel free to reach out, always interested in helping out others with their technical challenges :)

jackharrhy.dev

@jack@nfld.social

me@jackharrhy.com

slides will be available at
github.com/jackharrhy/why-the-web
sometime after the talk :)

again, thanks to everyone involved!

tools shoutout

EXCALIDRAW

Google Slides