Rike Exner

Frontend Technologien 2022: aktuelle Trends

XIMA DevDay

Hi!

Ich bin Rike Exner.

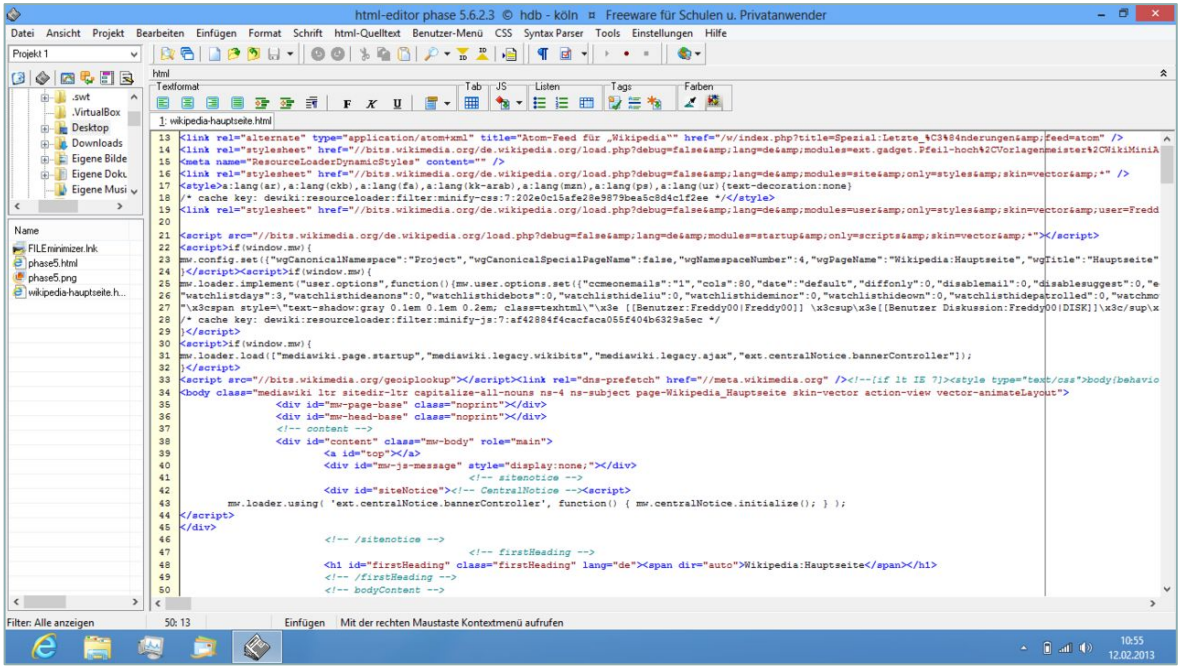
Frontend developer @IAV, Berlin
♡React ♡UI/UX ♡Web3

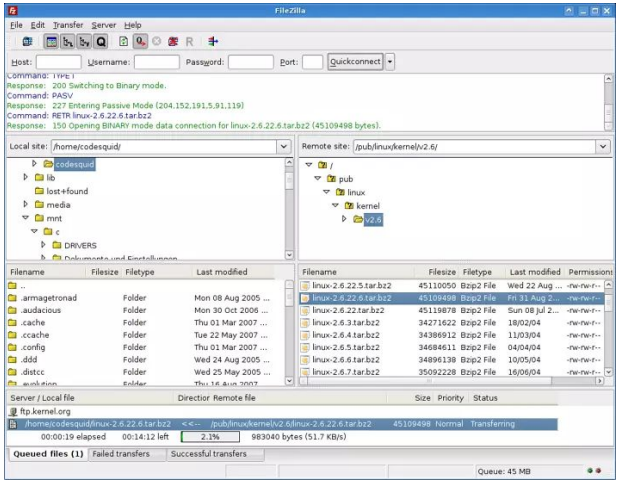
Twitter: @rikecodes

Slides: https://rike.dev/frontend-tech-2022.pdf

Web: rike.dev

LinkedIn: https://www.linkedin.com/in/rike-exner

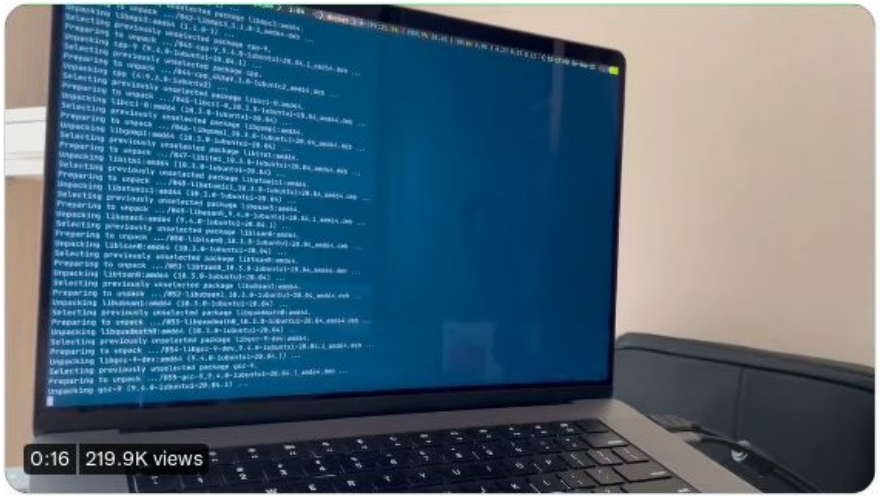




Web Developer Tech Stack 2002

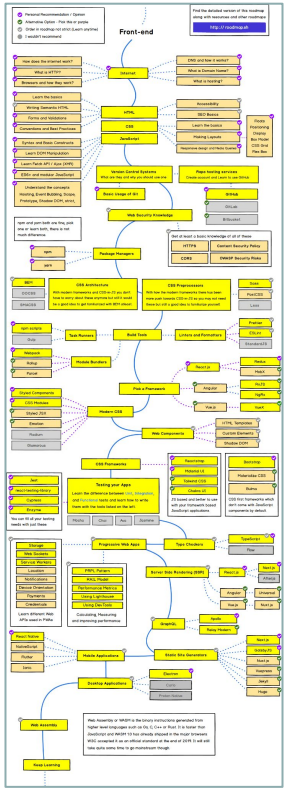
Tony Dinh 🎯
@tdinh_me

A frontend engineer compiling their “hello world” app in 2022:



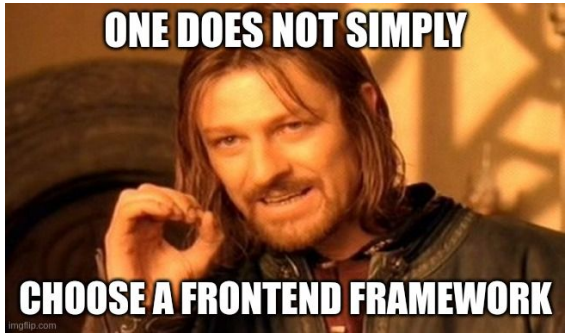
4:21 AM · Sep 26, 2022 · Twitter for iPhone

521 Retweets **79** Quote Tweets **6,535** Likes



Web Developer Tech Stack **2022**

Frontend Development in 2022 - quo vadis?



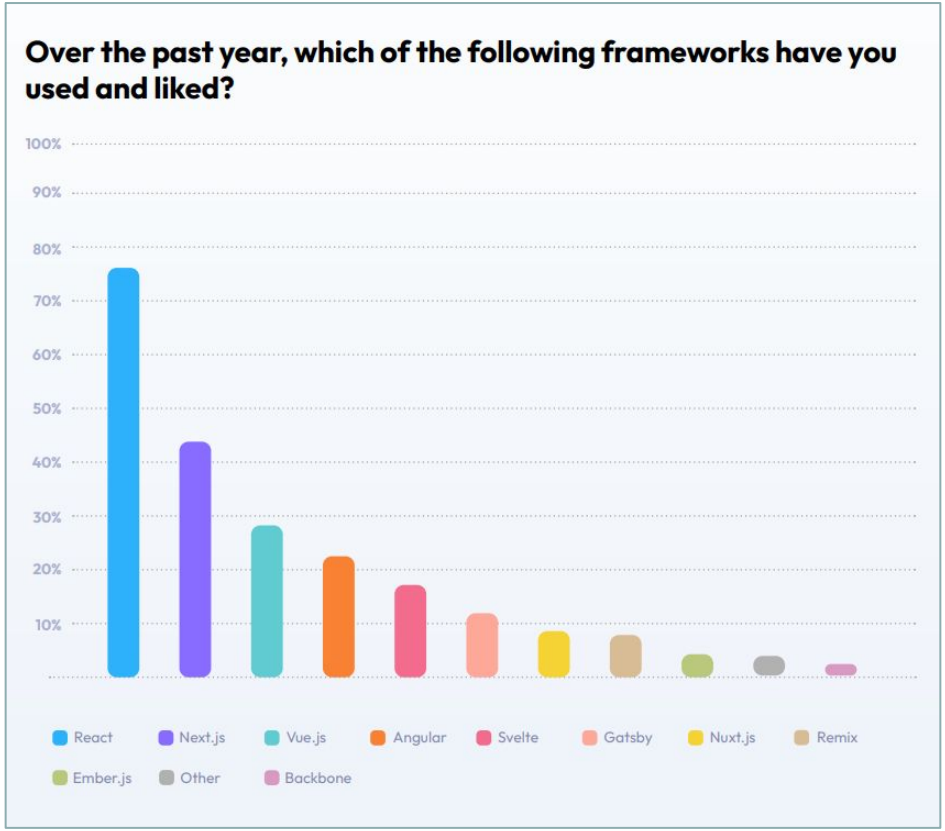
Verbreitung und Trends

Frontend Frameworks & Sprachen

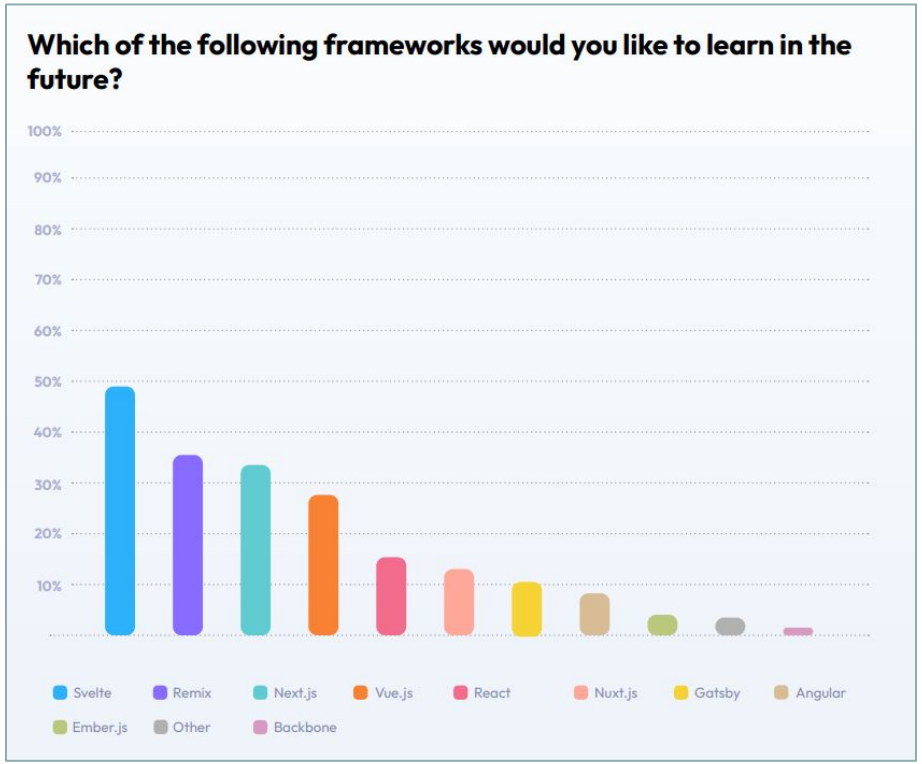
“

“new day, new JS framework”

~~“new day, new JS framework”~~

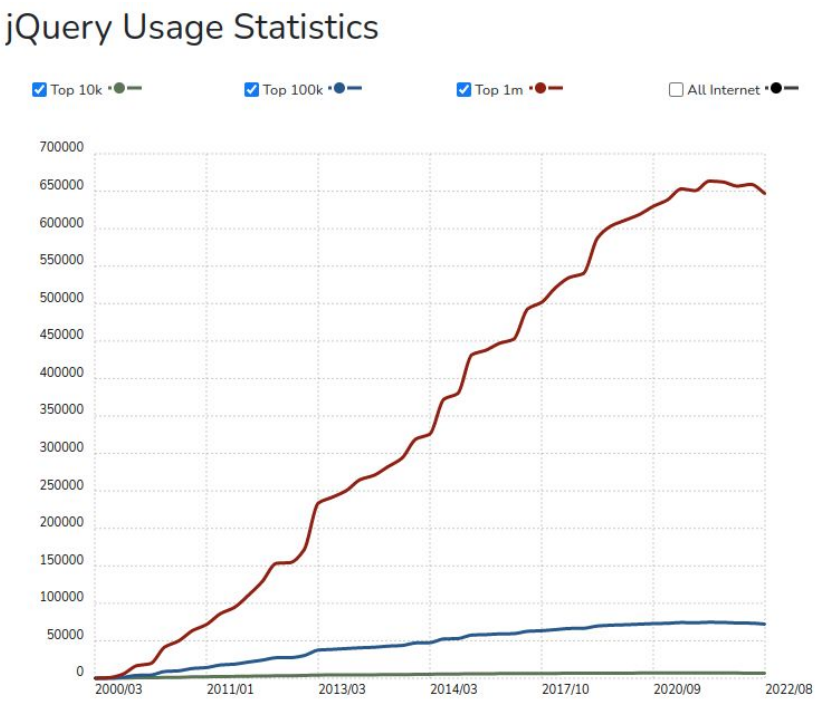


https://tsh.io/State-of-Frontend-2022.pdf



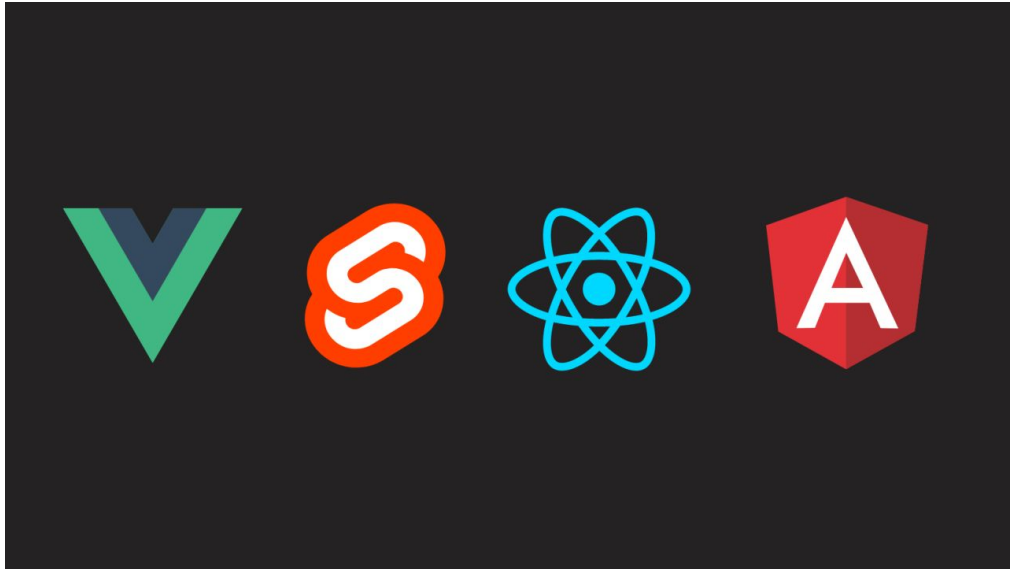
https://tsh.io/State-of-Frontend-2022.pdf

Einsatzzwecke, Vor- und Nachteile Frontend Frameworks



https://trends.builtwith.com/javascript/jQuery

React, Vue.js, Angular, Svelte



Gemeinsamkeiten

- Experten für Single-Page-Applications
- Open Source
- komponentenbasierte Architektur
- bringen Development Server / Hotreloads mit
- vsCode als perfekte Ergänzung

- *alle* sind performant..

Performance

- für den Praxiseinsatz sind Performance-Unterschiede im ms-Bereich und daher irrelevant

Name	svelte-v2.9.7-keyed	vue-v2.5.16-keyed	react-redux-v16.1.0 + 3.7.2-keyed	angular-ivy-v6.1.0-keyed
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	231.6 ± 3.3 (1.5)	158.8 ± 2.7 (1.0)	155.3 ± 1.4 (1.0)	173.8 ± 2.9 (1.1)
partial update Time to update the text of every 10th row (with 5 warmup iterations) for a table with 10k rows.	75.1 ± 3.8 (1.0)	156.4 ± 9.8 (2.2)	97.2 ± 5.2 (1.4)	71.6 ± 2.5 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	20.4 ± 4.4 (1.0)	20.0 ± 2.9 (1.0)	105.4 ± 2.7 (5.3)	108.8 ± 2.0 (5.4)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	354.4 ± 11.8 (1.3)	342.5 ± 6.0 (1.2)	274.1 ± 5.2 (1.0)	301.4 ± 65.1 (1.1)
clear rows Duration to clear the table filled with 10.000 rows.	183.5 ± 4.1 (1.1)	191.9 ± 6.1 (1.1)	173.9 ± 3.5 (1.0)	238.8 ± 3.6 (1.4)
slowdown geometric mean	**1.17**	**1.25**	**1.48**	**1.56**

https://cdn.accelebrate.com/images/blog/react-angular-vue-svelte-comparison/benchmark1.jpg

Also wo liegen denn dann die Unterschiede..?

“

React

- Leicht zu lernen
- Flexibel mit vielen Gestaltungsfreiheiten
- Hohe backward compatibility
- Große Community

- Architektonische Entscheidungen in großen Projekten z. T. schwierig, z. B. CSS handling, state management, testing

Vue

- Leicht zu lernen
- Sehr gute Dokumentation
- developer-friendly CLI
- Trennung von Struktur, Style & Funktion

- Kleinere Community

Angular

- Abgerundetes “Gesamtpaket”
- Forciert Nutzung von Typescript

- (über-) komplex
- Schwierig zu lernen

Svelte

- Bringt Implementierung von z. B. styles, animations & state management mit
- Leicht zu lernen, insbesondere mit nur HTML/CSS-Background

- (noch) sehr kleine Community
- Kleines Ökosystem

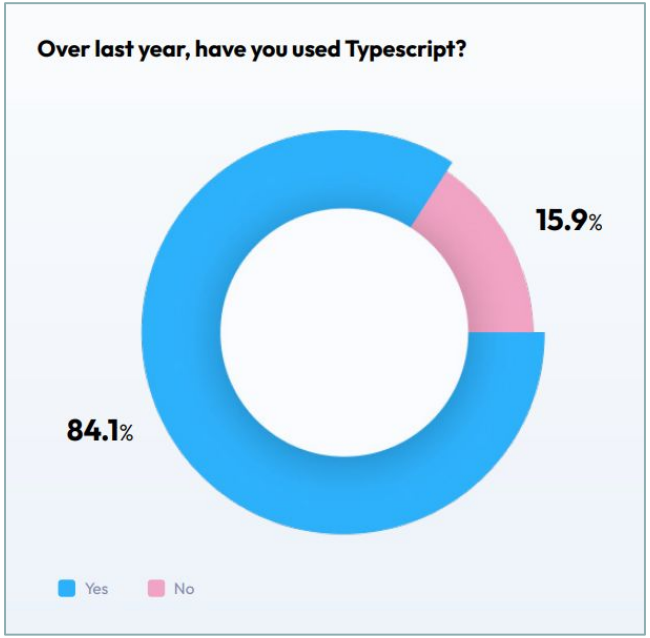
Welches Framework sollten wir für zukünftige Projekte nutzen?

“

Und sonst noch so..?

- Remix
- Astro
- Qwik
- Gustwind
- Solid

Typescript



https://tsh.io/State-of-Frontend-2022.pdf

Typescript Trends

- hier um zu bleiben: 85% aller Frontend-Entwickler*innen nutzten TS im vergangenen Jahr
- viele neue Libraries werden nativ in .ts geschrieben
- beeinflusst ECMAScript, aber ein “Ablösen” dessen ist nicht in Sicht

WebAssembly

WebAssembly - kommt das?

- Vorteile: hohe Performance und niedrige Dateigröße, wird bereits von allen großen Browserherstellern unterstützt
- sehr junge Community, die eher nicht in wirkliche Konkurrenz zu Javascript/Typescript treten wird
- eher ergänzend bspw. im Umfeld Gaming / VR

Design Systems / UI Libraries

“

CSS Handling

(React) CSS Handling Trends

- Vanilla CSS 👍 SASS/SCSS 👍 LESS 👎
- CSS Modules (?)
- JS-in-CSS (Styled Components, Compiled, Emotion, Goober, ..) (?)

Content-Management-Systeme und der Headless-Trend

“

Warum Headless?

- Headless CMS verzichten auf die Spezifizierung des Frontends und ermöglichen somit die Auslieferung der Inhalte auf verschiedenen Kanälen und Medien unabhängig von der finalen Darstellung
- *Prismic, Contentful, Strapi, Sanity, Storyblok, Directus*

Cross-Platform Development

Wild wild West..

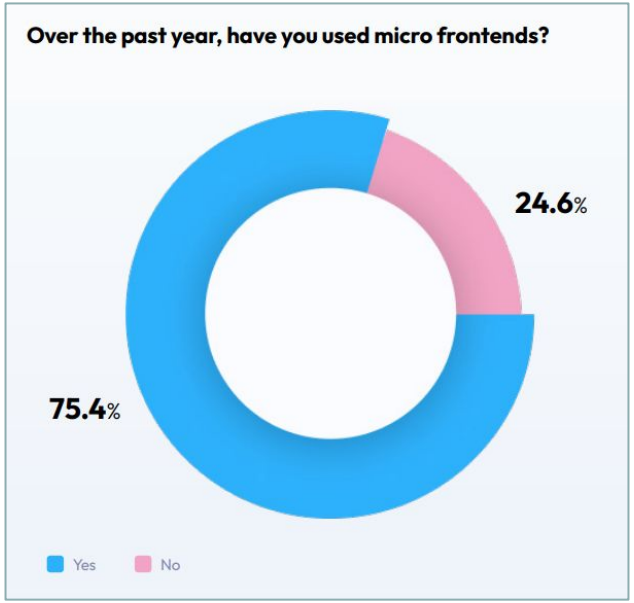
- Native App Development (Android/iOs) vs.
- Cross-Platform-Frameworks (z. B. Flutter, React Native, Kotlin Multiplatform Mobile, Xamarin, Ionic, Cordova, ..) vs.
- PWAs

Last but not least..

“

Microfrontends

- *Next.js with Module Federation: the Future of Micro Frontends?*
- *How Microfrontends Can Restructure Development Teams*



https://tsh.io/State-of-Frontend-2022.pdf

Accessibility

- *The A11Y Project*

Server Side Rendering, Static Site Generators & Static Site Hosting

- *Rendering on the Web*
- *Server Side Rendering (SSR) vs. Client Side Rendering (CSR) vs. Pre-Rendering using Static Site Generators (SSG) and client-side hydration*

Danke!
Fragen?

Slides:

https://rike.dev/frontend-tech-2022.pdf

Twitter: *@rikecodes*
Web: *rike.dev*
LinkedIn: *https://www.linkedin.com/in/rike-exner/*

Frontend Developer Trends

- Stackoverflow Trends: https://insights.stackoverflow.com/trends?tags=reactjs%2Cvue.js%2Cangular%2Csvelte%2Cangularjs%2Cvuejs3
- npm Downloads: https://npmtrends.com/@angular/core-vs-react-vs-svelte-vs-vue
- “The State of Frontend”: https://tsh.io/State-of-Frontend-2022.pdf?utm_campaign=SOFE2022&utm_medium=email&_hsmi=210998928&_hsenc=p2ANqtz-9LRvnvOEsj6YU49_IGfTjqkY4-gP8_mVnotPCIQXT8-Jf5bnu5hPppXBSZ09sn28Ysv8aR4TrCiYi1VoLA5qSrW7nJ4w&utm_content=210998928&utm_source=hs_automation
- Github stars statistics: https://star-history.com/#facebook/react&vuejs/vue&angular/angular&sveltejs/svelte&Date