

Bienvenue ! 🙌

| | Used and liked | Used and disliked | Want to learn in the future | Not interested | No opinion |
|------------|----------------|-------------------|-----------------------------|----------------|------------|
| Alpine.js | 7% | 1.9% | 9.2% | 35.6% | 46.3% |
| Angular2+ | 22.1% | 12.8% | 8.3% | 39.4% | 17.4% |
| Angular.js | 6.7% | 19.3% | 7.2% | 49% | 17.9% |
| Ember | 3.3% | 4.1% | 4.8% | 50.7% | 37.2% |
| HTMX | 7% | 2.5% | 36.4% | 27.3% | 26.7% |
| Lit | 6.5% | 2.5% | 14.4% | 32.2% | 44.5% |
| Phoenix | 1.8% | 0.5% | 9.8% | 34.2% | 53.7% |
| Preact | 9.7% | 3.6% | 16.5% | 35.4% | 34.8% |
| Qwik | 4.1% | 1.5% | 24.3% | 29.7% | 40.4% |
| React | 69.9% | 15.4% | 7.1% | 5.4% | 2.2% |
| Svelte | 25.8% | 3.5% | 43.6% | 12.9% | 14.2% |
| Vue.js | 44.8% | 8.6% | 24.4% | 13.1% | 9.1% |

You are
here

source: tsh.io/state-of-frontend

The Eternal Enemy: Complexity

apex predator of grug is complexity

complexity bad

say again:

complexity very bad

you say now:

complexity very, very bad

given choice between complexity or one on one against t-rex, grug take t-rex: at least grug see t-rex





htmx.org / CEO of Sun Microsystems (same thing)

@htmx_org

...

Unbothered. Moisturized. Happy. In My Lane. Focused. Garbage
Collecting.

[Traduire le post](#)



Stéphane Trébel @StephaneTrebel · 15h

C'est la semaine prochaine ! 😊

Vendredi en huit je serai au @jugsummercamp de 16h25 à 17h10 en salle
Casoar Tadorne pour causer de @htmx_org en revenant sur le "tout JavaScript",
ses limites, et ce que nous apporter cette librairie...

- ❗ Un constat sur le dev Web "moderne"
- 👵 HTMX, un retour aux sources
- ✊ Alors, Révolution ou pas Révolution ?

**LES LIMITES DU « TOUJOURS
PLUS DE JS ! »**

LOOK AT ME

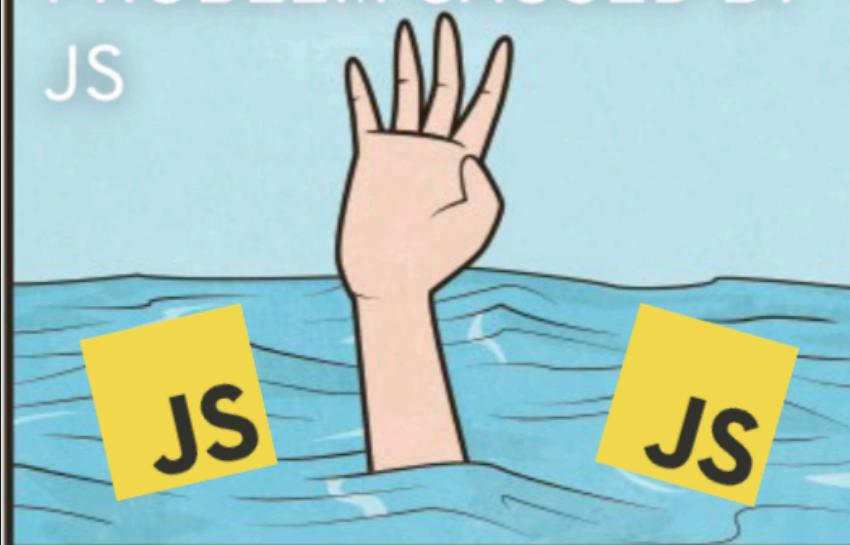
WEB DEVELOPMENT

I'M YOUR PLATFORM, NOW

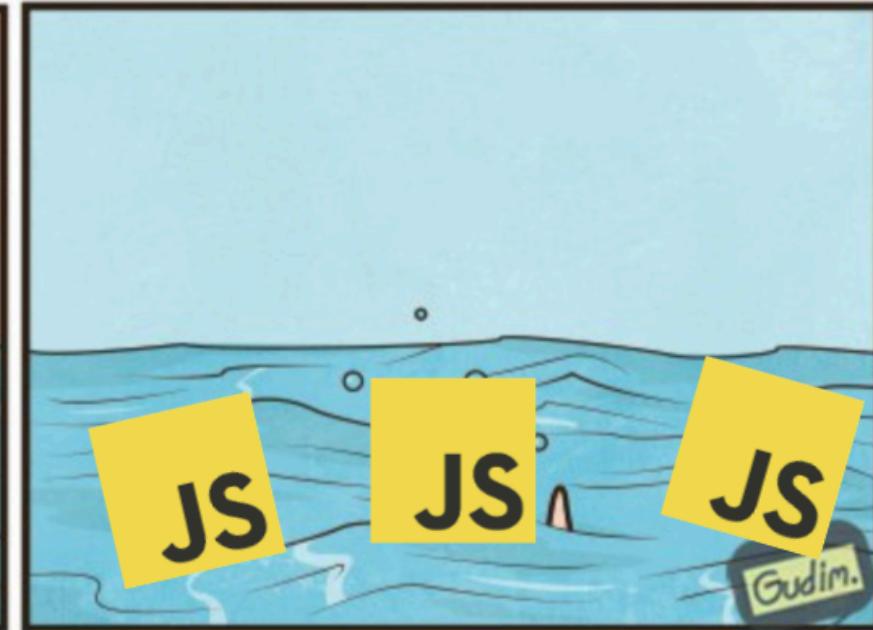
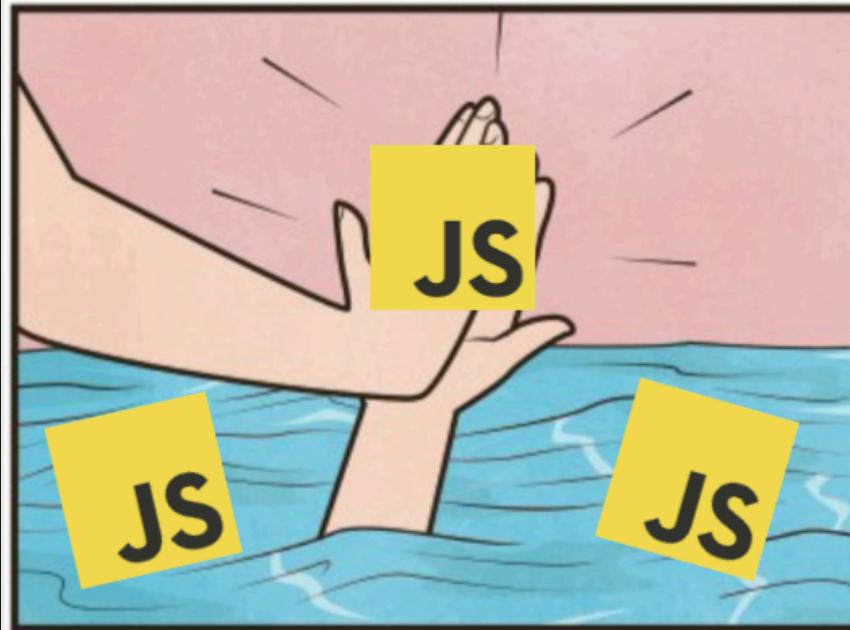
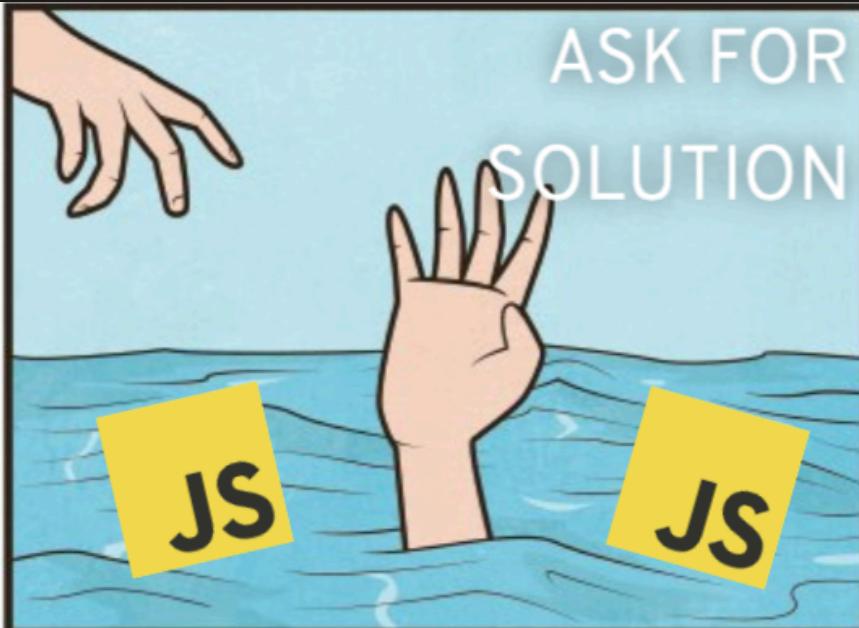
imgflip.com

À la base, ça n'existe pas, une "application Web".

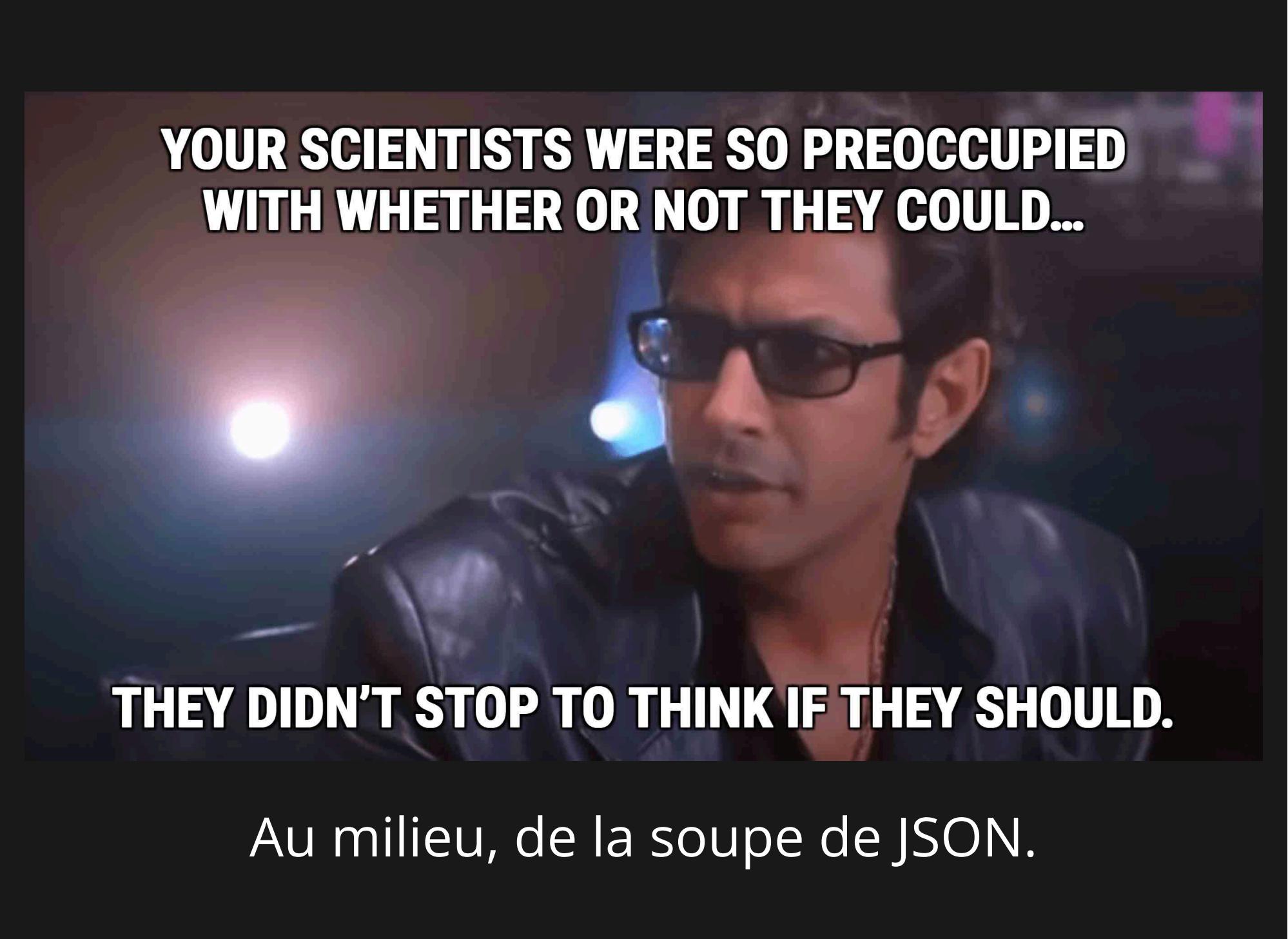
PROBLEM CAUSED BY
JS



ASK FOR
SOLUTION



Réinventer la roue, constamment.



**YOUR SCIENTISTS WERE SO PREOCCUPIED
WITH WHETHER OR NOT THEY COULD...**

THEY DIDN'T STOP TO THINK IF THEY SHOULD.

Au milieu, de la soupe de JSON.

YOU WENT FULL-REACT



imgflip.com

La fameuse "Javascript fatigue"...



Recherche



Accueil



Réseau



Offres d'emploi



Messagerie



Notifications



Vous ▾



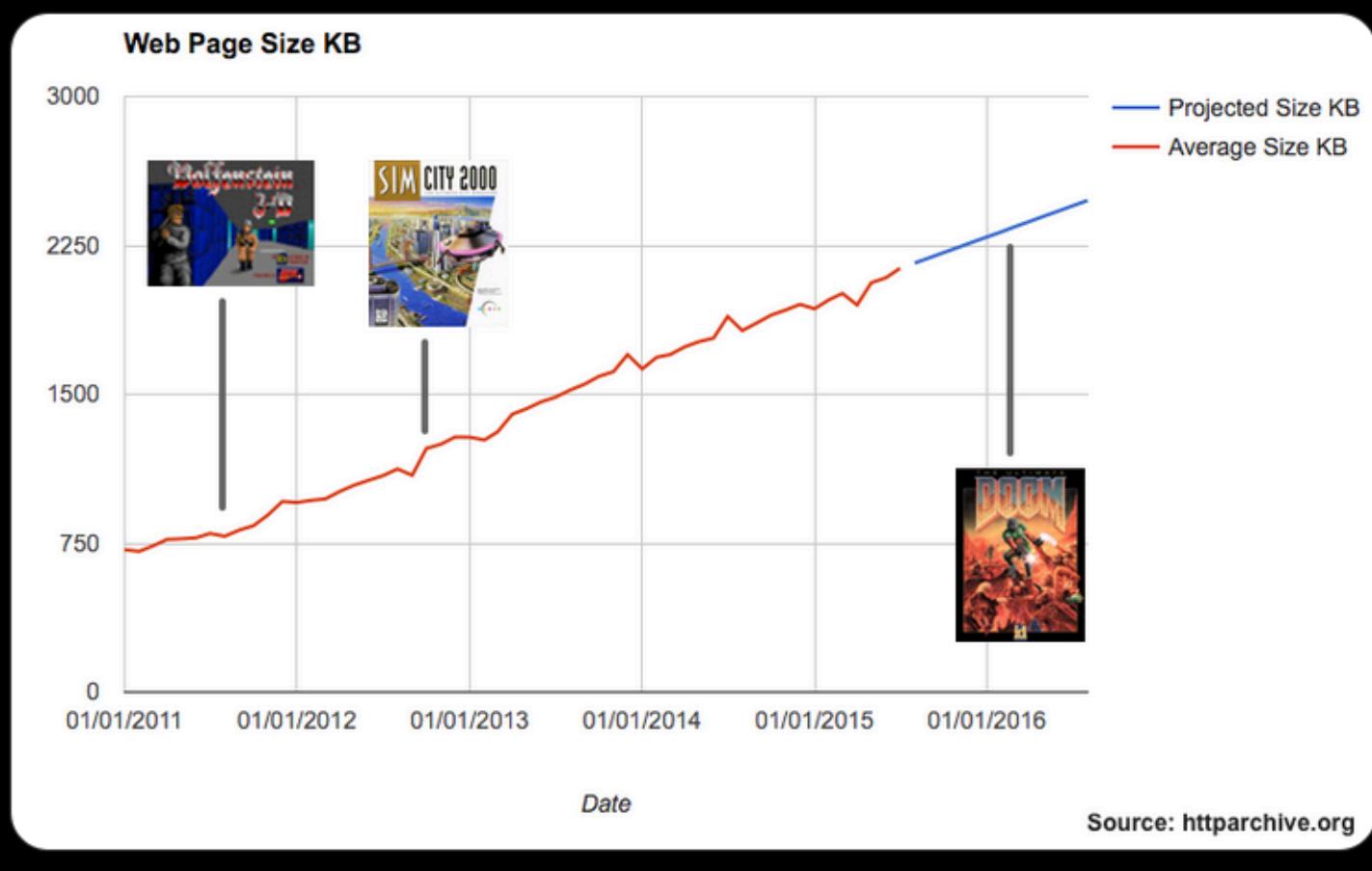
ronan cremin
@xbs

...

In about 7 months average web page size will be same as Doom install image.

Well done us! Onwards & upwards!

[Traduire le post](#)



Le principe d'erooM

On pardonnera d'imaginer le principe d'erooM — "*Effort Radicalement Organisé d'Optimisation en Masse*" — qui voudrait fixer comme objectif le fait d'arriver à optimiser le logiciel d'un facteur deux tous les deux ans. En quelque sorte, d'augmenter la vitesse d'exécution comme le faisait la loi de Moore en son temps, mais cette fois-ci sans devoir changer tout le matériel à chaque génération. Cela permettrait de continuer à inventer de nouveaux usages aux ressources numériques ainsi libérées, tout en consommant moins de ressources, moins d'énergie, tout en continuant à fournir du travail à l'industrie du logiciel.

Un autre monde est possible 😊

PARLONS DONC D'HTMX



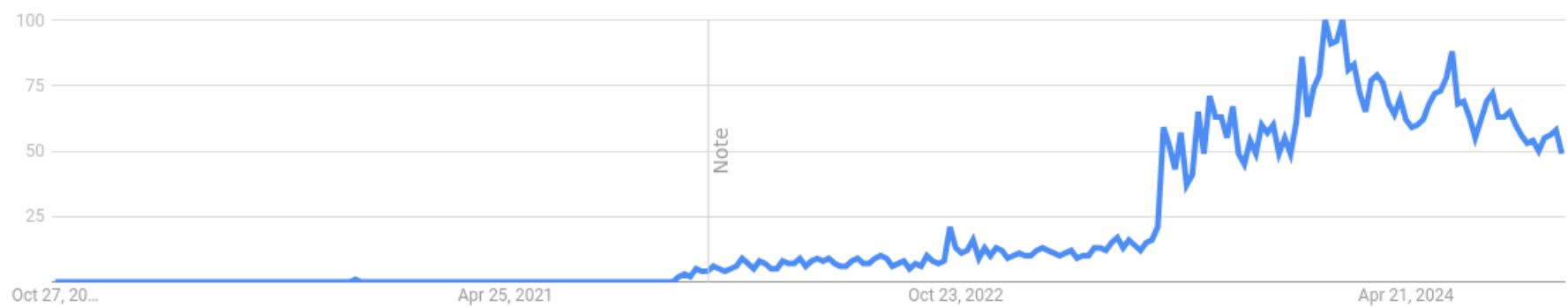
imgflip.com



No module
bundlers
at all

Interest over time ?

⬇️ ⏴ ⚡



HATEOAS et AJAX



(non ce ne sont pas des héros grecs)

```
<button hx-post="/clicked" hx-swap="outerHTML">  
  Click Me  
</button>
```

“Quand un utilisateur clique sur ce bouton, envoie une requête AJAX vers /clicked, et remplace le bouton par le HTML de la réponse.”

```
<table>
  <thead> ... </thead>
  <tbody id="result"> ... </tbody>
</table>
<form
  method="POST"
  hx-post="/post-form"
  hx-target="#result"
  hx-swap="beforeend"
>
```

“Quand un utilisateur soumet le formulaire, envoie une requête AJAX POST vers /post-form, et ajoute le HTML de la réponse juste avant la fin du tag ayant l'id result.”

```
<tr hx-get="/contacts/?page=2"
    hx-trigger="revealed"
    hx-swap="afterend">
  <td>Agent Smith</td>
  <td>void29@null.org</td>
  <td>55F49448C0</td>
</tr>
```

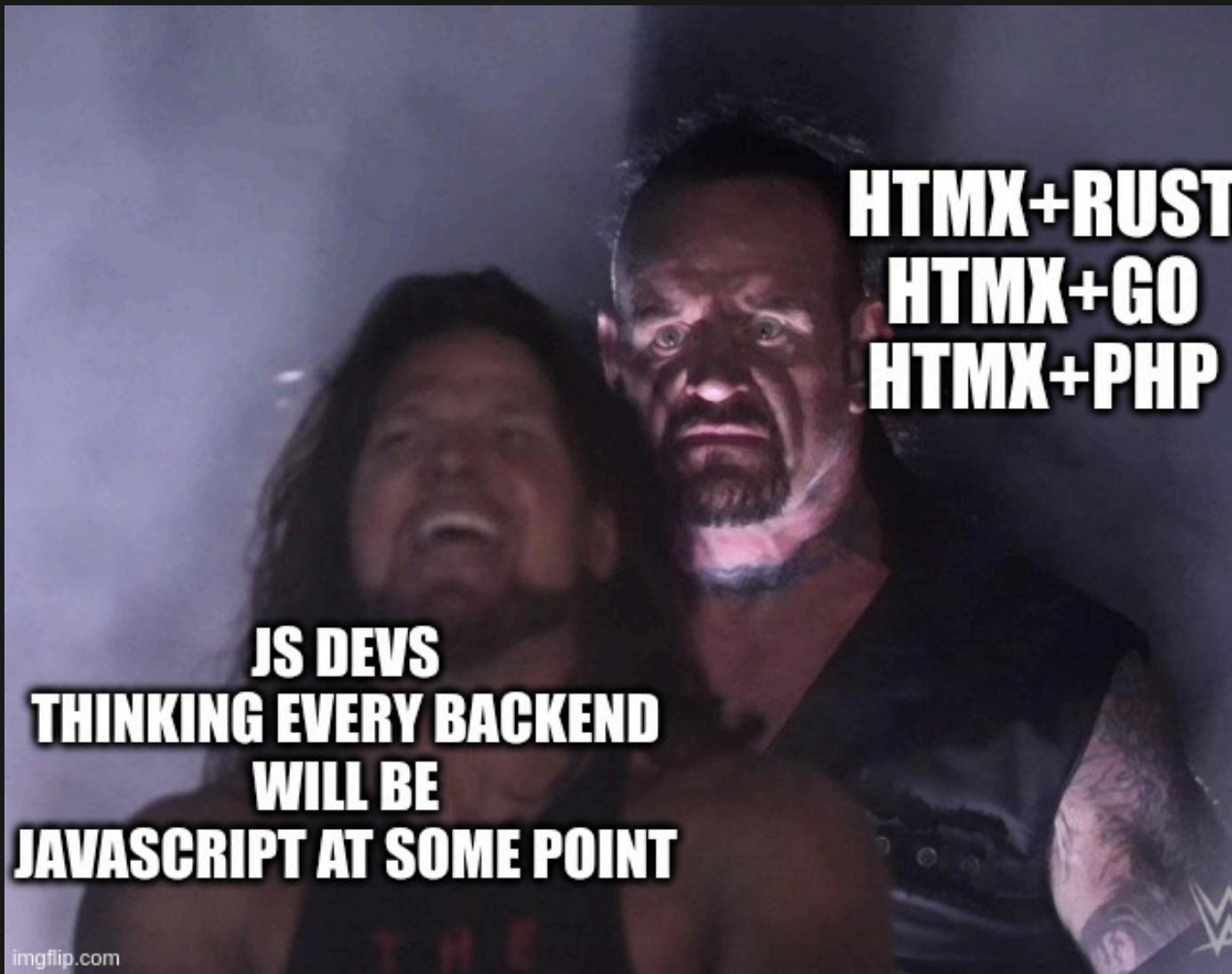
“Lorsque cette ligne est révélée dans le viewport, envoie une requête GET vers /contacts/?page=2, et ajoute le HTML de la réponse après cette ligne”

Oh vous voulez plus complexe ? Ok !

```
<h2>Contacts</h2>
<table class="table">
  <thead>...</thead>
  <tbody id="contacts-table" hx-get="/contacts/table"
    hx-trigger="newContact from:body"> ... </tbody>
</table>
<h2>Add A Contact</h2>
<form hx-post="/contacts"> ... </form>
```

*“Lorsqu'on soumet le formulaire,
envoie une requête POST vers
/contacts. La réponse aura un
en-tête HX-Trigger: newContact
qui déclenchera une requête GET
vers /contacts/table, ce qui
rafraîchira la table.*

UN CH'TI POC



- 🎨 Front en HTML (incluant des WebComponents), du CSS, et...HTMX !
- ⌚ Juste du JS dans les WebComponents pour leur cycle de vie
- 🦀 Un backend en...Rust !

C'est parti !



EN GROS, C'EST PHP/DJANGO/ROR ?



lol you are going to use
twenty year old technology
to build modern websites
lol maybe for a toy website
good luck w/ that lmao



Yes.



no. sixty years old

**"BORING TECHNOLOGIES",
FOR LACK OF A BETTER WORD**

ARE GOOD

NE PLUS PENSER EN
FRONT/BACK MAIS EN
FRONT+BACK

You are a developer, you bring a solution

« Front+Back y a trop de choses à savoir ! »

En résumé:

- Front + Back sont l'«application» Web
- HTML d'abord
- CSS pour décorer
- JS *uniquement* quand c'est nécessaire

En résumé:

- Front + Back sont l'«application» Web
- HTML d'abord
- CSS pour décorer
- JS *uniquement* quand c'est nécessaire

C'est tout.

En résumé:

- Front + Back sont l'«application» Web
- HTML d'abord
- CSS pour décorer
- JS *uniquement* quand c'est nécessaire

C'est tout.

Le reste, c'est votre boulot.

HTMX est ce que le HTML aurait dû devenir
avant qu'on bascule dans le «tout Javascript»

POURQUOI HTMX EN 2025 ?

- 💪 WebComponents sont (enfin) matures
- ✍ CSS a beaucoup évolué
- ⭐ Javascript a également évolué

*“Il ne faut toucher à l'architecture
que d'une main tremblante.”*

Montesquieu contemporain

POUR ALLER PLUS LOIN



10 Tips For Building SSR/HDA applications

Carson Gross

June 13, 2022

Building web applications using traditional Server-Side Rendering (SSR) or, saying the same thing another way, building [Hypermedia-Driven Applications](#) (HDAs) requires a mindset shift when compared with building web applications with Single Page Application frameworks like React.

If you come at this style of development with an SPA-engineering hat on, you are likely to be frustrated and miss out on many advantages of this particular architectural choice.

Here are 10 tip to help you make the mental shift smoothly, taking advantage of the strengths of this approach and minimizing the weaknesses of it:

Tip 1: Maximize Your Server-Side Strengths

A big advantage of the hypermedia-driven approach is that it makes the server-side environment far more important when building your web application. Rather than simply producing JSON, your back end is an integral component in the user experience of your web application.

- 🚫 Une critique assez forte du «Javascript EVERYWHERE»
- 👩‍💻 Gérons le métier avant de se faire plaisir avec de la technique
- 🍰 À chaque couche (HTML, CSS, JS,) son objectif, ses affordances, etc.

Vous n'êtes pas Dev X/Y/Z.



Vous n'êtes pas Dev X/Y/Z.



Vous êtes un Dev —tout court—



Qui suis-je ? 🤝





Les liens



Vos feedbacks



@StephaneTrebel

HTMX est un retour aux sources du Web, de ce qui le rendait fun, et il est prêt pour le futur ! 😊