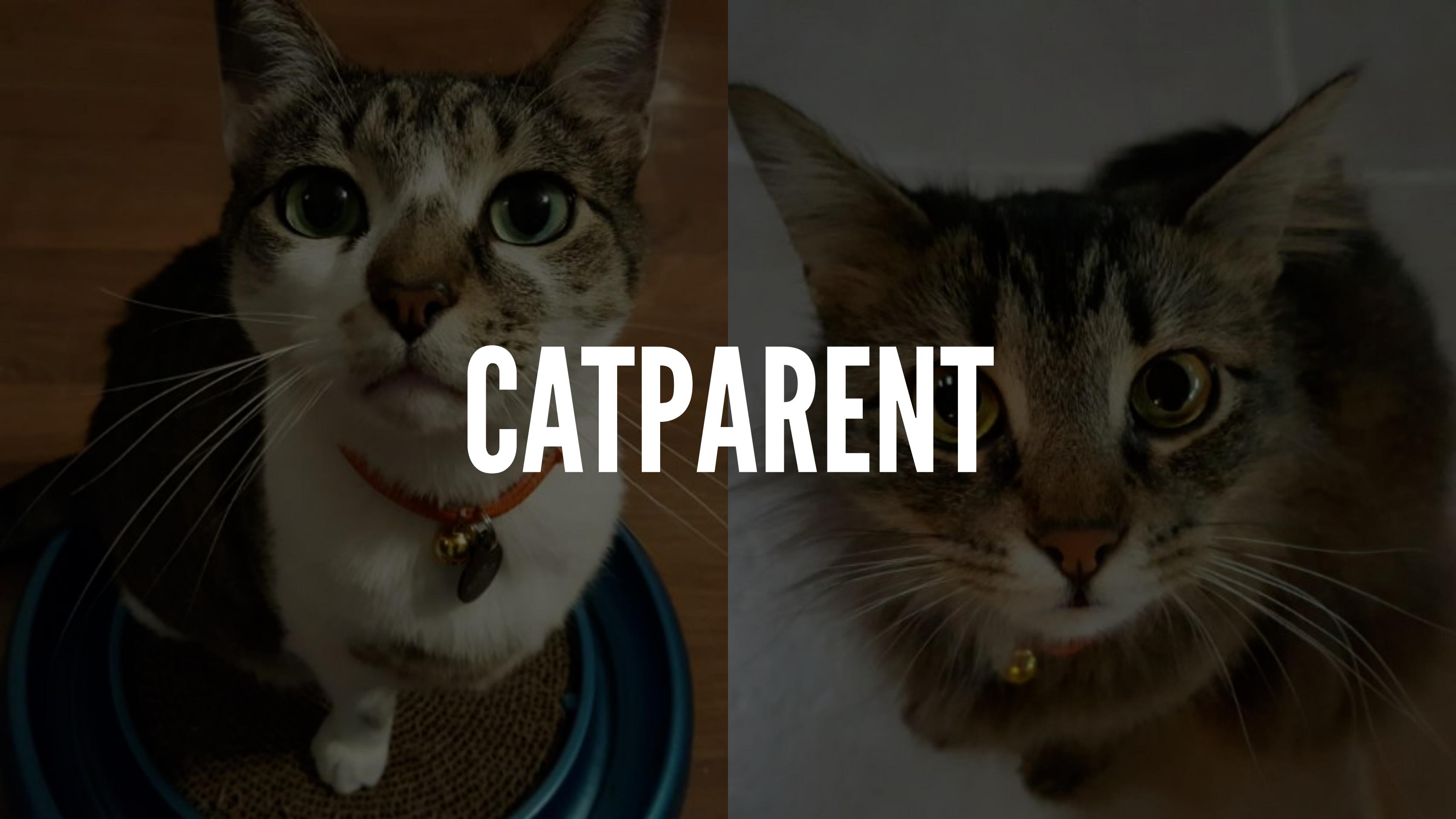


V8, WEBASSEMBLY, AND THE *Future of JS and a Multi- Language Web*



ABOUT ME

- ▶ Robotics Author/Addict
- ▶ Developer 
- ▶ Twitch streams hardware/software
@nodebotanist



CATPARENT

WHAT EVEN IS WebAssembly?

WHAT WEBASSEMBLY IS NOT

- ▶ a programming language
- ▶ the death of JS
- ▶ something you can just ignore cause it's gonna go away



WHAT WEBASSEMBLY IS

- ▶ A compilation target for other languages to compile to
 - ▶ An augmentation of the abilities of JS
 - ▶ But most importantly...

► Pretty literally* magic



* - no not really literally but I'll explain later

WEBASSMEBLY IS A COMPIRATION TARGET

- ▶ You write code in other languages and compile them into WebAssembly
- ▶ Rust, C/C++, Go, C#; these are just a few of the languages with WebAssembly as a compile target

**BUT THE QUESTION YOU THEN
HAVE TO ASK IS...**

WHAT

npm





**THERE ARE SO MANY
REASONS YOU WOULD WANT
THIS IN YOUR LIFE**

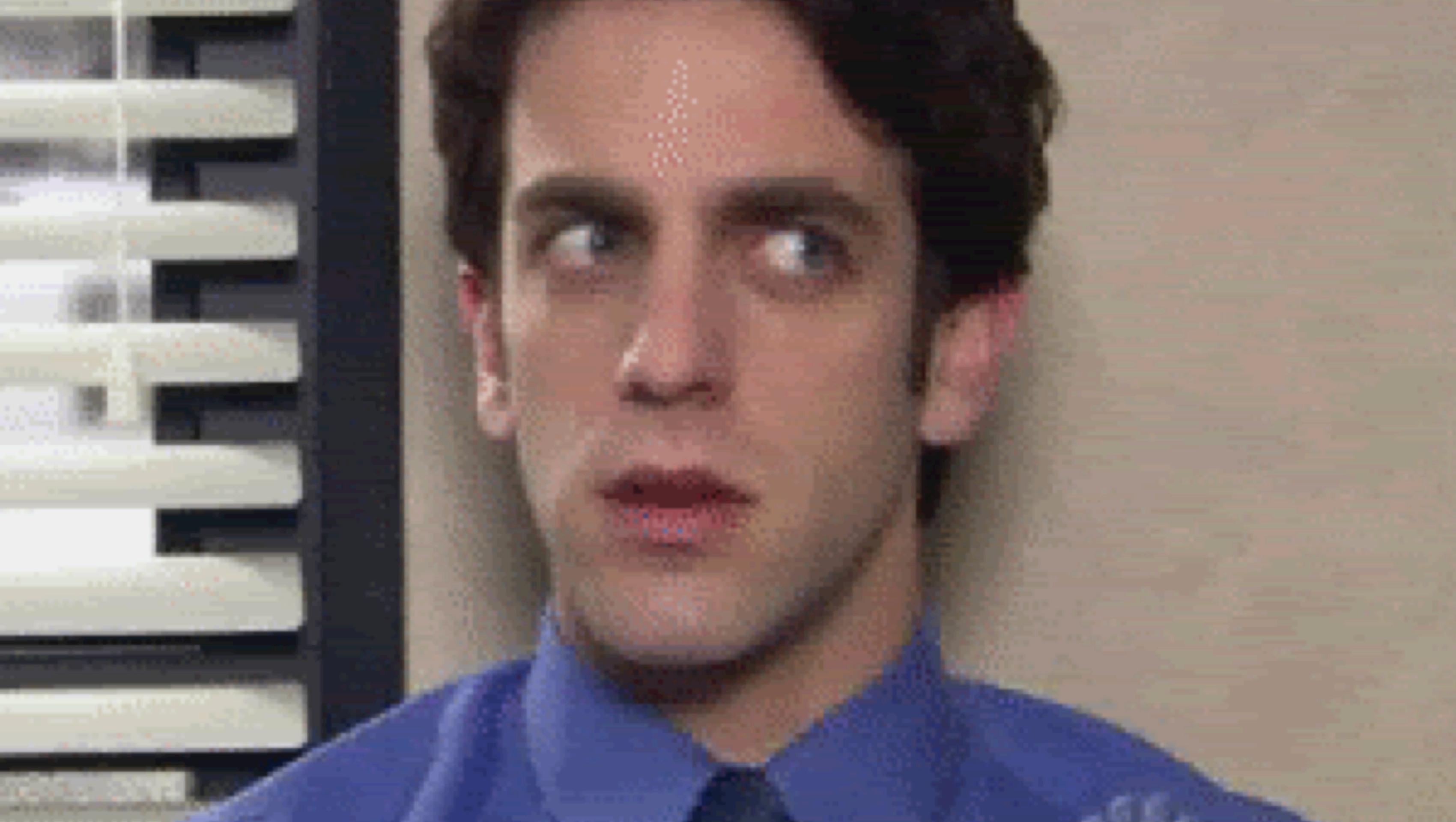
THIS IS THE NEXT
AJAX FOR THE
BROWSER



NO BUT SERIOUSLY

Ajax lent a huge credibility to browser applications, and WebAssembly
is an evolution of that.





Browser

HTML

CSS

"JavaScript"

Server

click link

return page

click link

return page

LITERALLY
EVERYTHIN
OVER
HERE

Browser

HTML

CSS

JavaScript

With AJAX

Server

request page

get page

AJAX Request

Result

AJAX

result

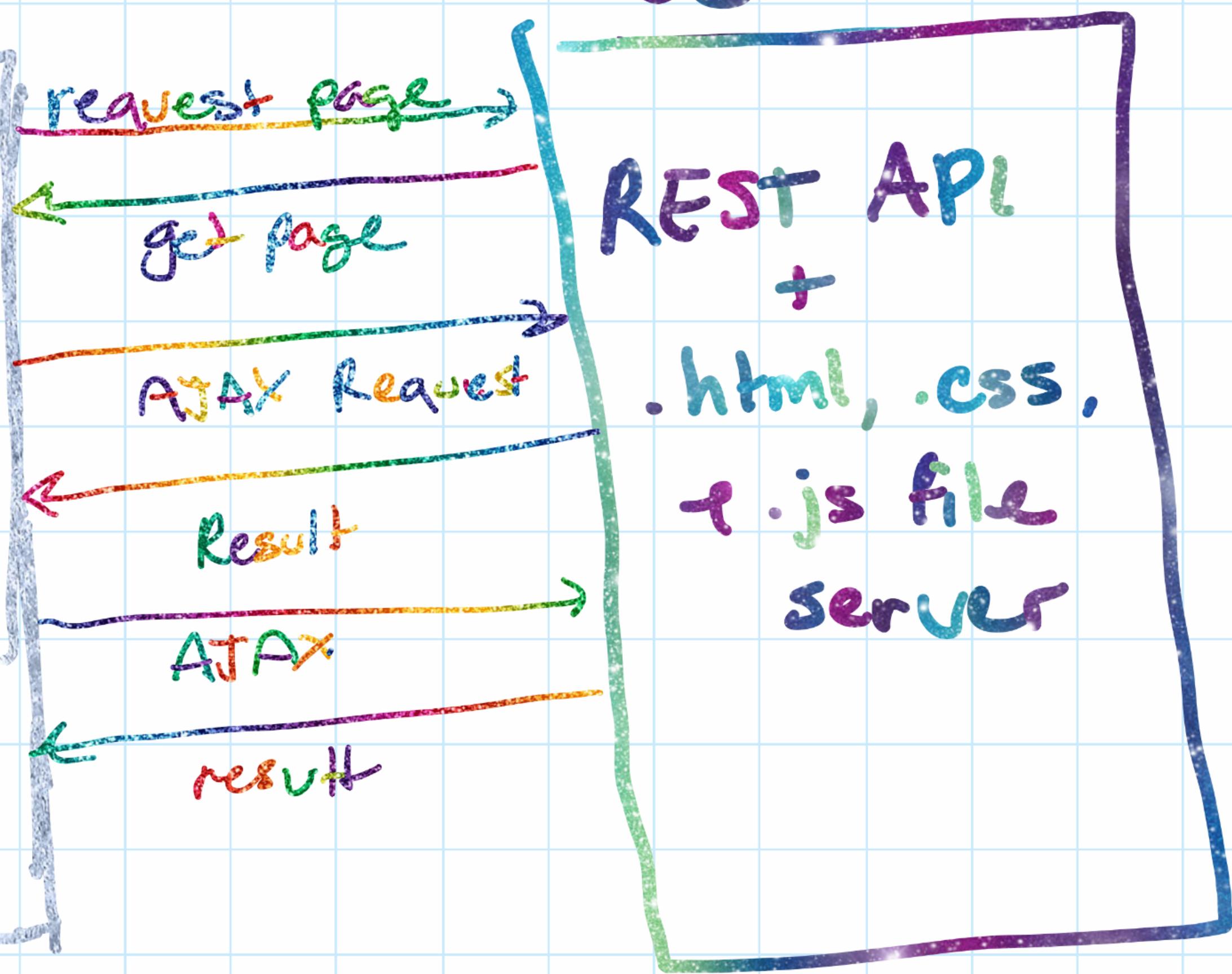
REST API

+

.html, .css,

+ .js file

server



Browser

Server

HTML

CSS

JavaScript

With AJAX
and Service
Workers

And WebAssembly

request page

return page

use WebAssembly
to Compute

use service worker
3rd party module

AJAX

yeah OK.

LOL OK.



WHY DOES THIS MATTER?

- ▶ Augmenting JS at its not-so-strong points
- ▶ Not rewriting entire codebases to use them on the web
- ▶ Fewer calls to the server, less latency, faster web apps

AUGMENTING JS AT ITS NOT-SO-STRONG POINTS

```
0.1 + 0.2 === 0.3 //false
```

If you're running anything that relies on mathematical numerical accuracy that meant, until now, another AJAX call to have another language do alllllllll the math. With WebAssembly, we can do this in the browser, with, say, Rust.

OTHER JS NOT-SO-STRONG POINTS

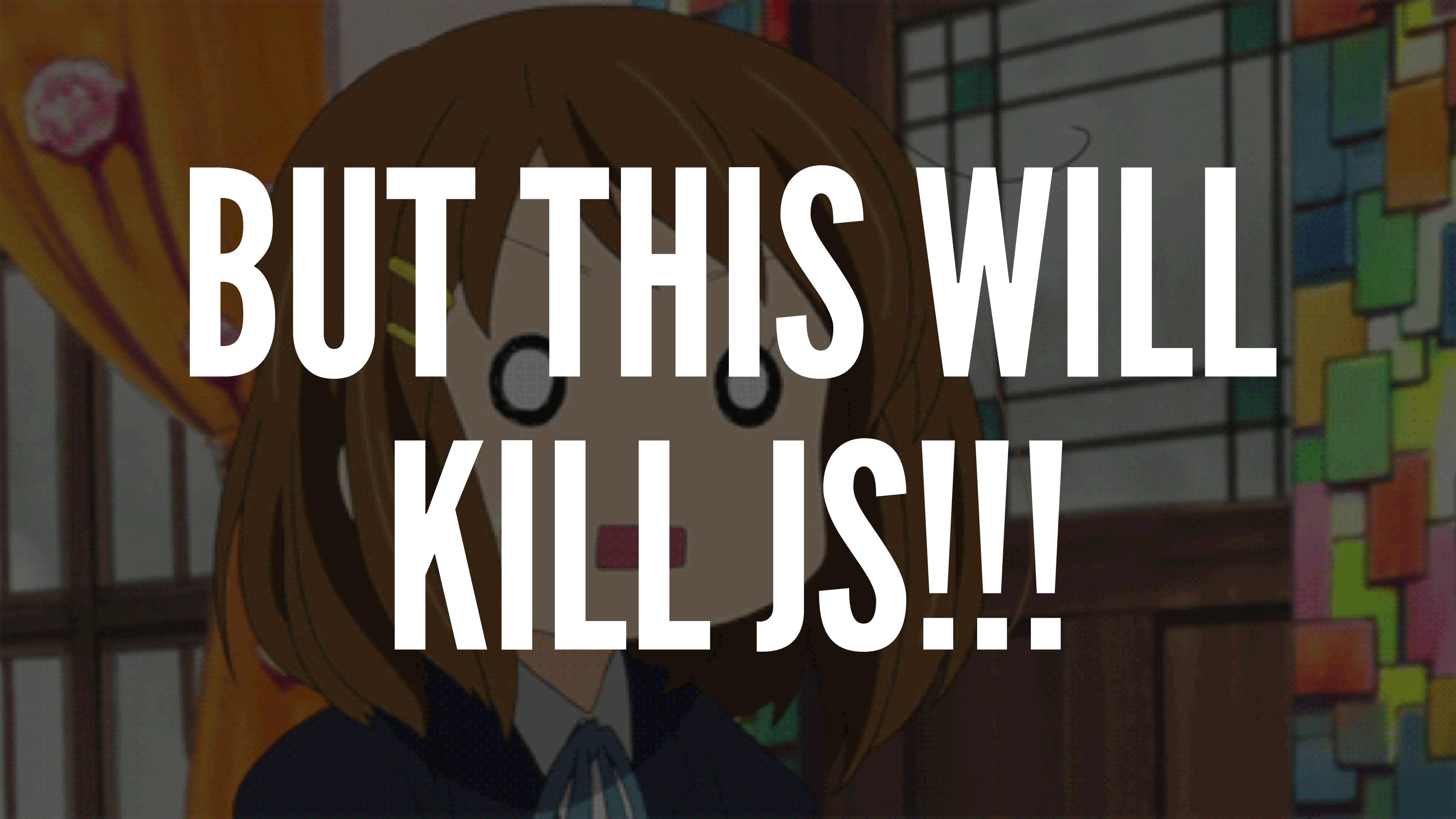
- ▶ Type coercion side-effects: "" == 0 //true
 - ▶ API Overloads:

```
1 + document.getElementById("inputElem").value; // Concatenates  
1 + Number(document.getElementById("inputElem").value); // Adds
```

- ▶ The mystery that is typeof

```
typeof {} === "object" //true  
typeof "" === "string" //true  
typeof [] === "array"; //false
```

USING WEBASSEMBLY MEANS
USING THE RIGHT TOOL FOR
THE JOB



**BUT THIS WILL
KILL JS!!!**

**NO, IT MAKES JS BETTER BY LETTING IT DO
WHAT IT IS GOOD AT AND IGNORING THE
REST.**

IT MAKES THE WEB BETTER BY CREATING BETTER BROWSER EXPERIENCES



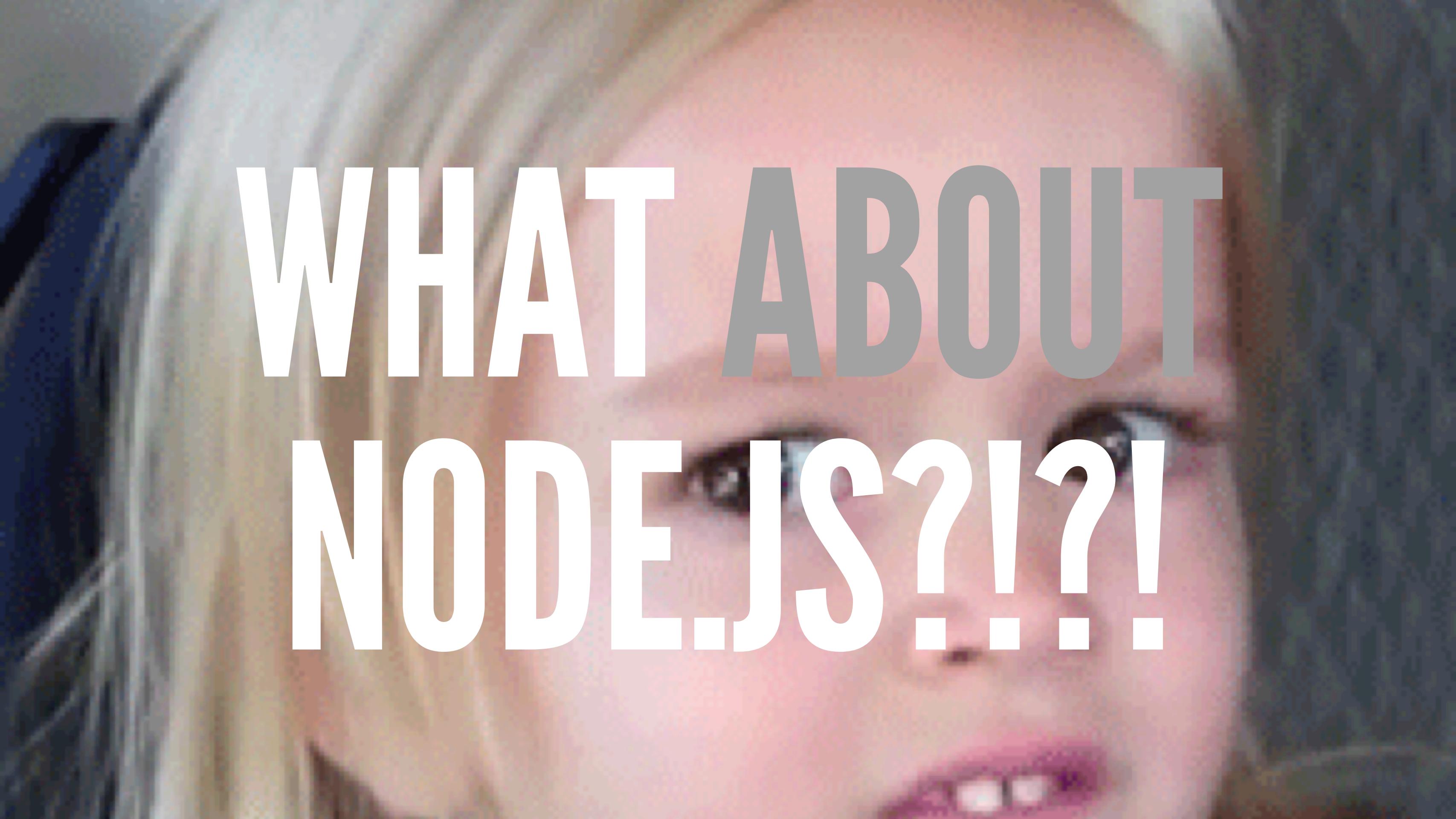
**LET'S TAKE A
CLOSER LOOK
WITH A DEMO**

THE DEMO

- ▶ uses wasm-imagemagick
- ▶ manipulates images in the browser up to 10x faster than JS can
- ▶ Shows the real power of not having to rewrite code and being able to let us use the right tool for the job

**BUT WHAT ABOUT
NODE.JS?**

WAIT ...



**WHAT ABOUT
NODE.JS?!?!**

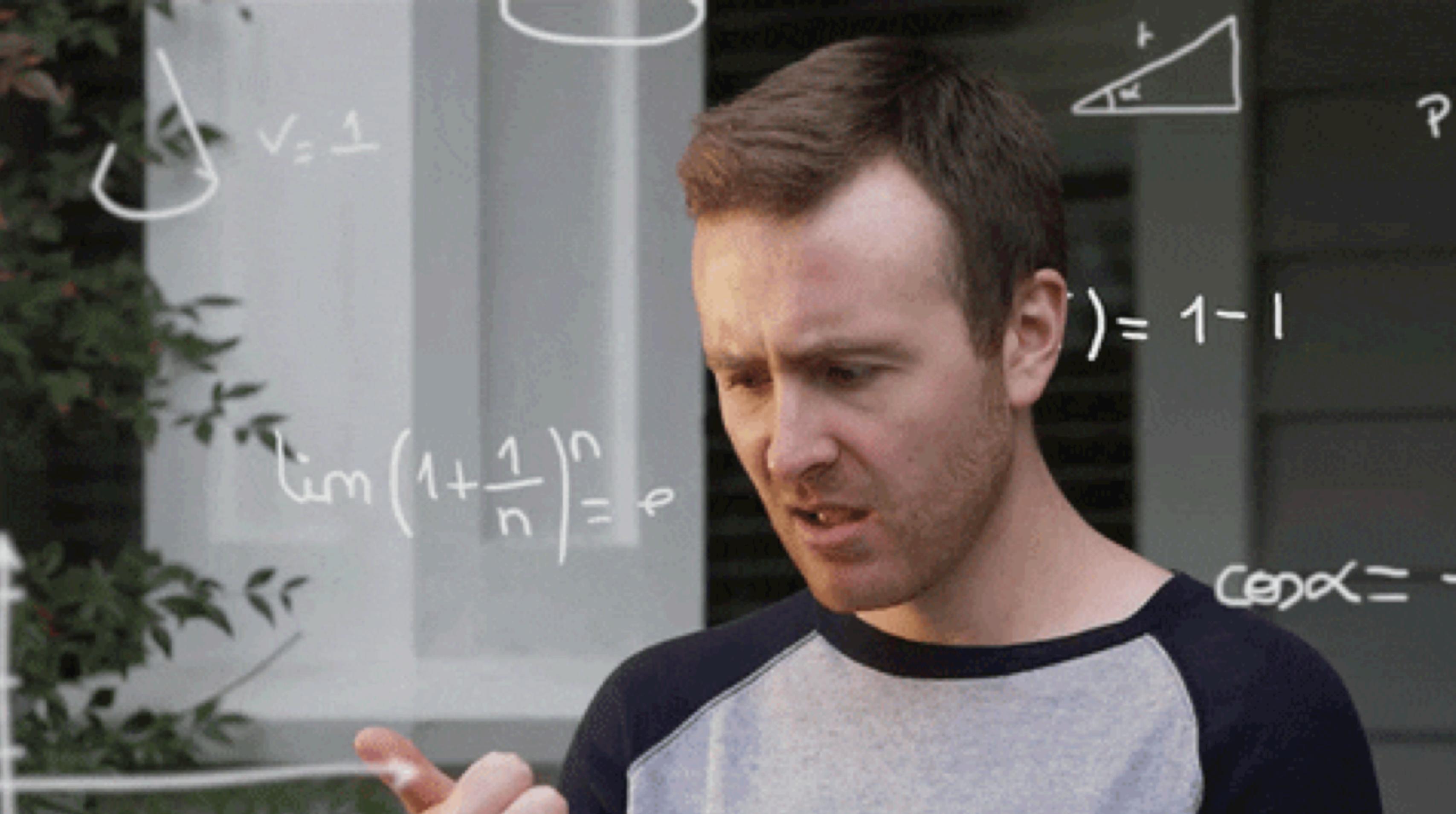
NATIVE. HECCIN. MODULES.



WHY NATIVE MODULES ARE SUCH A PAIN

- ▶ They have to be recompiled on download
- ▶ They either have to compile on every platform OR leave off platforms from support
- ▶ Node-Gyp (disclaimer: I respecc the hecc out of their work.)

WEBASSEMBLY
WORKS ON NODE
>= 8.0



WEBASSEMBLY MODULES IN NODE.JS

- ▶ Are precompiled for NodeJS, so they're portable to any platform that runs NodeJS.
- ▶ No more recompilation on every download on every architecture.
 - ▶ FOR REALS.

"Everyone wants to [deprecate] node-gyp and WebAssembly would allow us to do this"

– Laurie Voss, this morning

WEBASSEMBLY IS EVEN INVASING SERVERLESS



CLOUDFLARE®
Workers

WE HAVE A FREE TIER NOW!



(I'll show this again later and it's at our booth)

THE POINT OF THIS TALK

- ▶ Try WebAssembly (I personally really like Rust)
- ▶ WebAssembly is the future of JS in all its forms
- ▶ If you are a hiring manager; hire someone who is different from you.
Just go and do it.



THANKS FOR LISTENING!



- ▶ kas@cloudflare.com
- ▶ [@nodebotanist](https://twitter.com/nodebotanist)
- ▶ Search [meetup.com](https://www.meetup.com) for 'Serverless' for a meetup on Tues here in Singapore where you can learn to write a worker

