WHY SYNTAX CHANGES MATTER (A.K.A WHY CHANGES TO JS MEAN A LOT MORE THAN YOU THINK)

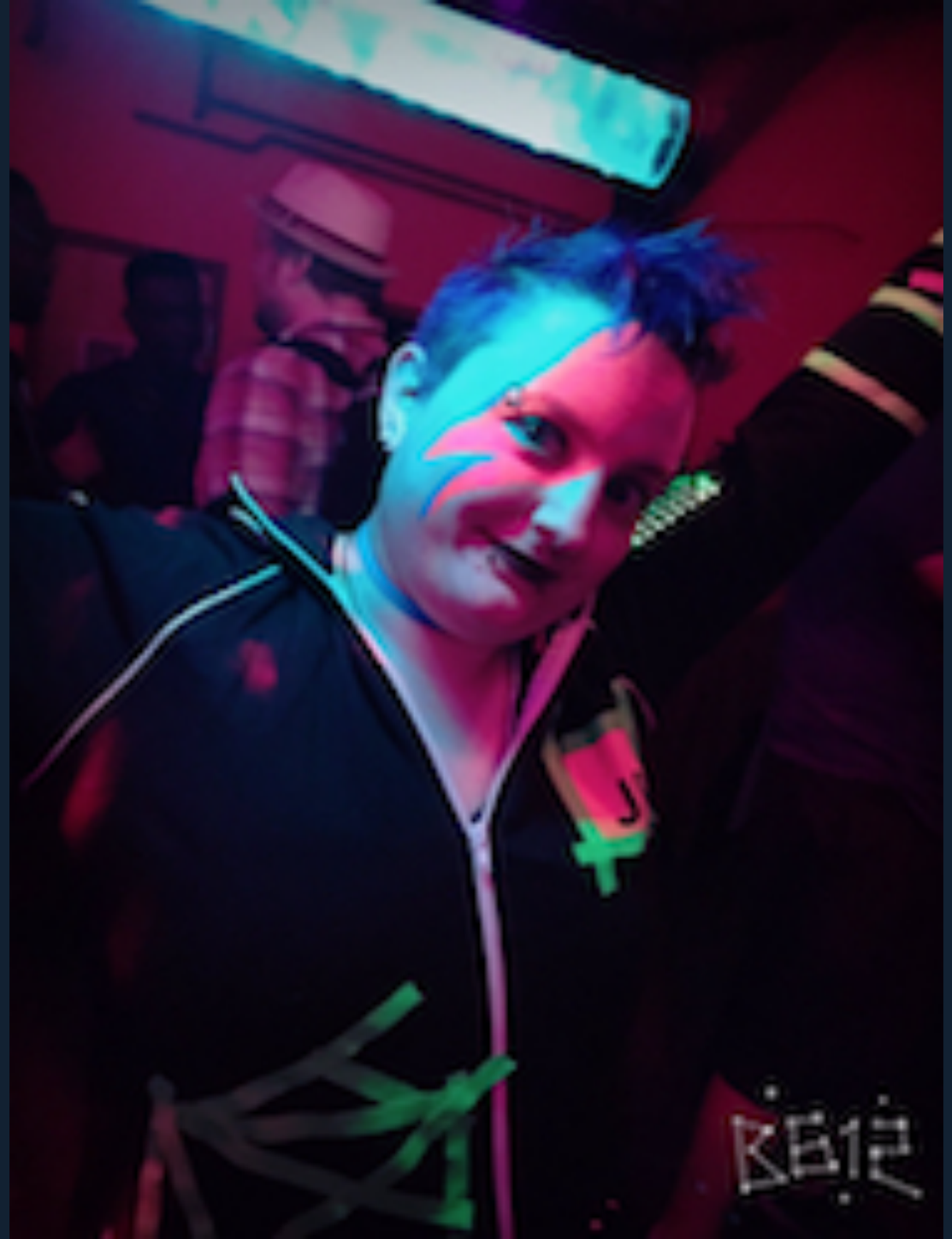
HEY! I'M KAS

» Developer Relations Engineer at Auth0/ Webtask.io

» Serverless/Node.JS afficionado

» Also pretty good at Auth (OAuth, OIDC, etc.)

» Gender non-binary (they/ them, she is OK, Yes you may ask polite questions)



I'M ALSO A HARDWARE HACKER

» Some people actually just call me 'nodebotanist'

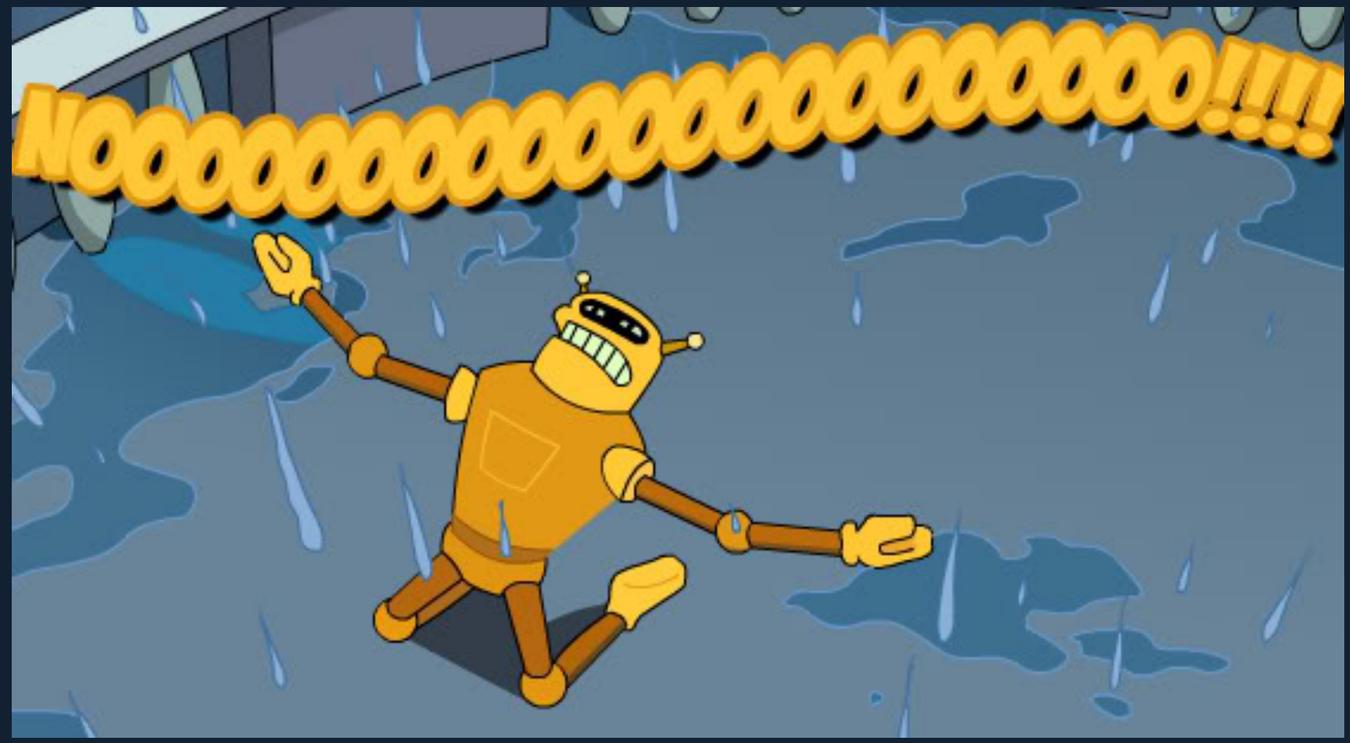
» In-Progress EE degree

» Tessel open-source committee member

» I wrote a book on JS Robotics

HANG ON, MY DRESS WANTS TO SAY "HI!".

BUT TODAY I'M NOT HERE TO TALK ABOUT THAT



(But come ask questions later!)

TODAY I'M HERE TO TALK ABOUT EDUCATION

"BUT YOUR TALK TITLE SAYS..."

I KNOW WHAT IT SAYS. BUT THEY'RE RELATED.

CLOSE YOUR EYES FOR A SECOND, AND PRETEND YOU'RE LEARNING SOFTWARE DEVELOPMENT FOR THE FIRST TIME

WHAT DOES THIS DO?

```
function foo(x=4){
  return () => {
    let sum = x--;
    for(y = x; y > 0; y--){
      sum *= y;
    }
    return sum;
  }
}

console.log(foo(5)() + foo()());
```

IF YOUR RESPONSE STARTS WITH "WELL, OBVIOUSLY..."

... you didn't pretend hard enough (or didn't try)

IF YOUR RESPONSE WAS MORE...

Good! You're ready!

IF YOUR RESPONSE WAS MORE...

"Well [expletive], even without pretending I don't get it! I'm a terrible programmer!"

CALM DOWN. NO YOU'RE NOT.

You are smart and awesome and cool, don't let overly clever code get you down.

"OVERLY CLEVER CODE"? DON'T WE WANT CODE TO BE AS CLEVER AS POSSIBLE?

WE WANT CODE TO BE WELL-WRITTEN.

And clever use of abstractions is a part of that.

The thing is, good code is a balance of a lot of different things that are subjective and situation-dependent.

LIKE WHAT?

» Are you working on a team? What in your team composition?

» Are you teaching someone with this (the answer is almost always yes!)?

» Are you open sourcing this? Do you want contributors?

» What's your primary function? What can you not do without?

BALANCING THAT SOUNDS HARD

THE SECRET OF SOFTWARE DEVELOPMENT

Writing code isn't the hard part. Solving problems in a way that is sustainable for your team, completes its function, using the right tools is the hard part.

SO HOW DOES THIS RELATE TO ES 2015 AND NEW YEARLY JS RELEASES?

(I say ES 2015 because I want to pretend 2016 didn't happen. Not ES 2016, just 2016 as a year. Unless the Cubs win the world series. Then maybe.)

ES6 ADDED A LOT OF AWESOME ABSTRACTIONS FOR PREVIOUSLY ARCANE/ANNOYING FEATURES OF JS:

» Arrow syntax for callbacks

» let for block syntax

» rest and spread for optional arguments and array manipulation

» generators

» Destructuring for moving values and other fun

AND NOW WE'RE GETTING YEARLY RELEASES FOR STANDARDS!

BUT THESE ABSTRACTIONS MAKE A MUCH STEEPER LEARNING CURVE-- NOT JUST FOR NEW PROGRAMMERS, BUT THOSE OF US TRYING TO KEEP UP!

Software developers exist on a constant state of needing to learn new things. So becoming a software developer can be like trying to learn how a car works while it moves at about 5 mph. If you're already a developer, it's 2 mph.

SOME ABSTRACTIONS MAKE CODE EASIER TO READ. SOME MAKE IT HARDER

» Class syntax is usually easier to teach new devs.

» Destructuring...not so much.

THE PROBLEM IS WE HAVE TO TEACH (OR LEARN) THE HARD WAY TO DO THINGS, USUALLY BEFORE WE TEACH/ LEARN THE EASY WAY

CLASS SYNTAX

learning the theory behind classes is a big stepping stone. Trying to wedge how JS prototypes into that is a nightmare.

But once you get how classes work, you have to follow through with how prototypes work in JS.

DESTRUCTURING

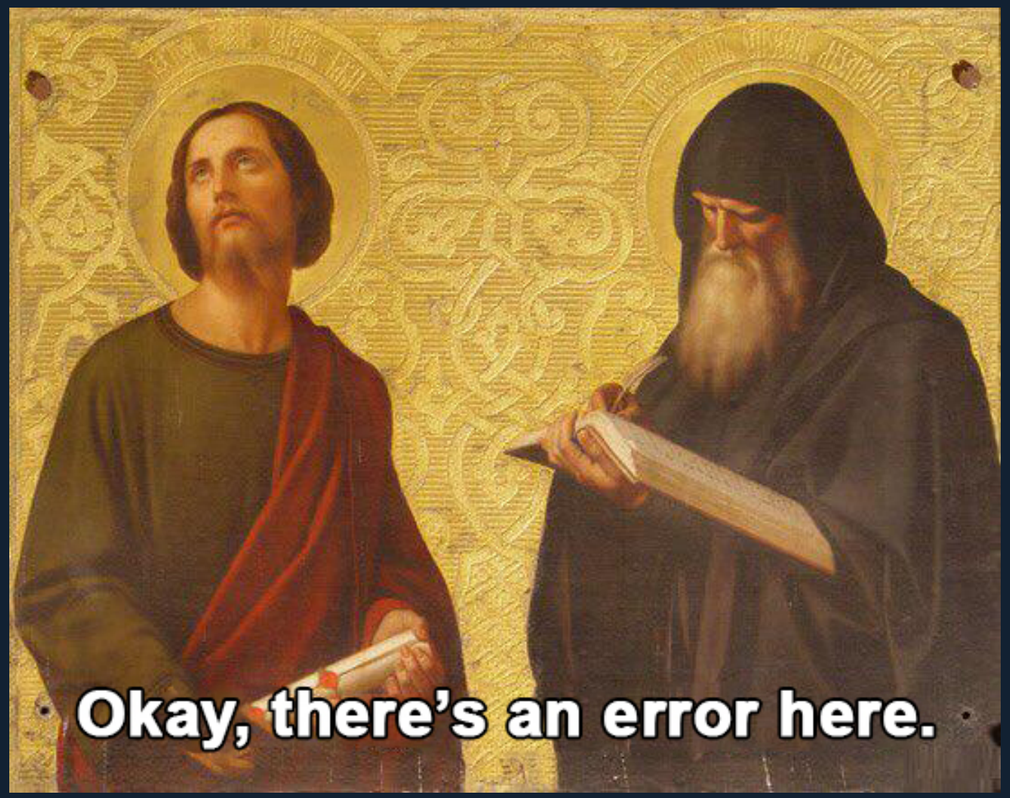
Arrays are tricky, but easier to grasp when you use temp variables and do things the verbose way first.

Destructuring makes sense when you learn how arrays work!

SO IT TURNS OUT EVERY TIME YOUR MATH TEACHER MADE YOU DO THINGS THE HARD WAY, THEN TAUGHT YOU A FORMULA? THAT WAS FOR YOUR OWN GOOD.

AN ASIDE ON CODE REVIEWS

We need to change the way we think about code reviews from this:



AND THIS

TO THIS

HOW?

» Create a code standard

» Publicize it

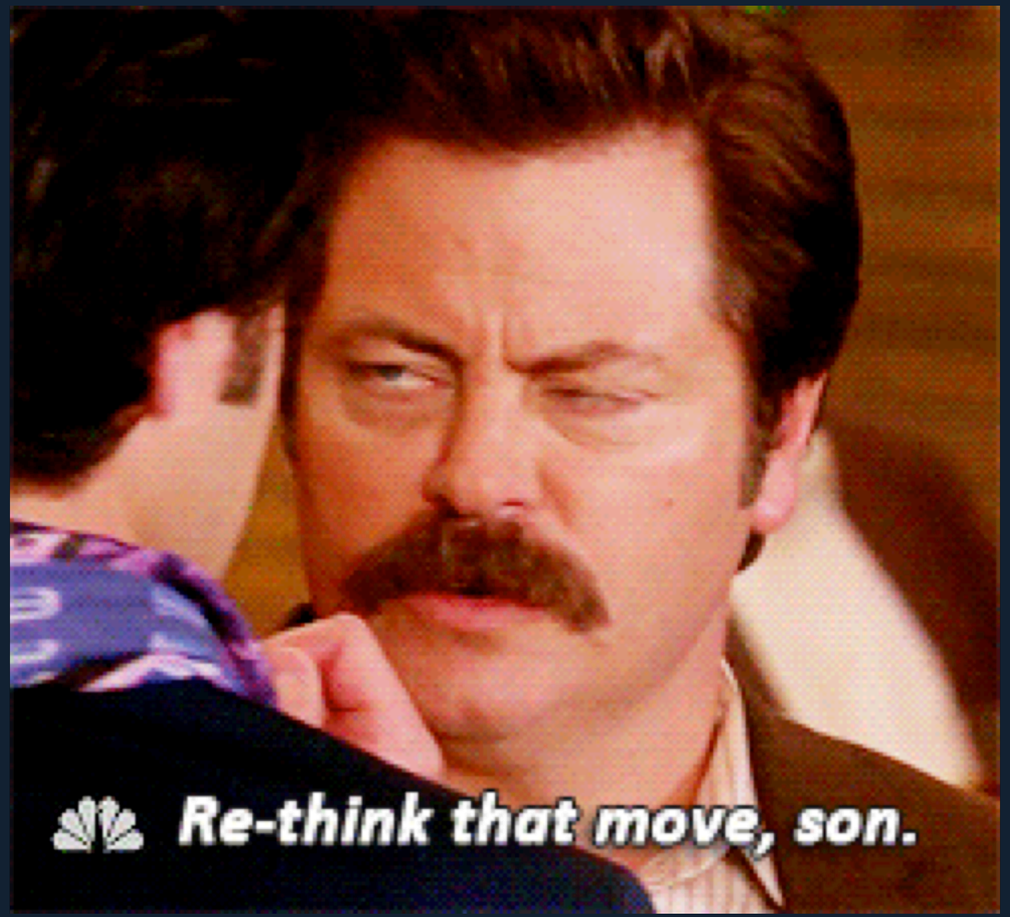
» Use the tools (JSHint, ESLint, whatever!)

» Code reviews as a team exercise

» Encourage mentoring and create learning opportunities

BUT WHY DO WE CARE ABOUT NEW DEVELOPERS?

(If you're really asking yourself that...you might want to think about your outlook and get yourself some perpsective...)



NEW DEVELOPERS ARE VITAL TO THE CONTINUATION AND LIFE OF THE SOFTWARE DEVELOPMENT COMMUNITY

THE PROBLEM WITH PROBLEMS

We don't solve some really important problems because we don't have the perspective to be aware they exist or solve them in a way that actually helps those that are affected

MORE DEVELOPERS =
MORE DIVERSITY =
MORE PERSPECTIVE =
BETTER SOLUTIONS.

SO LET'S DO OUR PART

We need to think more about how to make our code more readable, more understandable, and reach out to new developers.

IT'S HARD. BUT IT'S ABSOLUTELY WORTH IT.

IN SUMMARY

» ES2015 and new features are great!

» But we need to keep new (and not-so-new) folks in mind!

» We're all in this together

THANKS FOR LISTENING!

@nodebotanist on twitter, github (slides are up there under frontporch-2016)
the@nodebotani.st