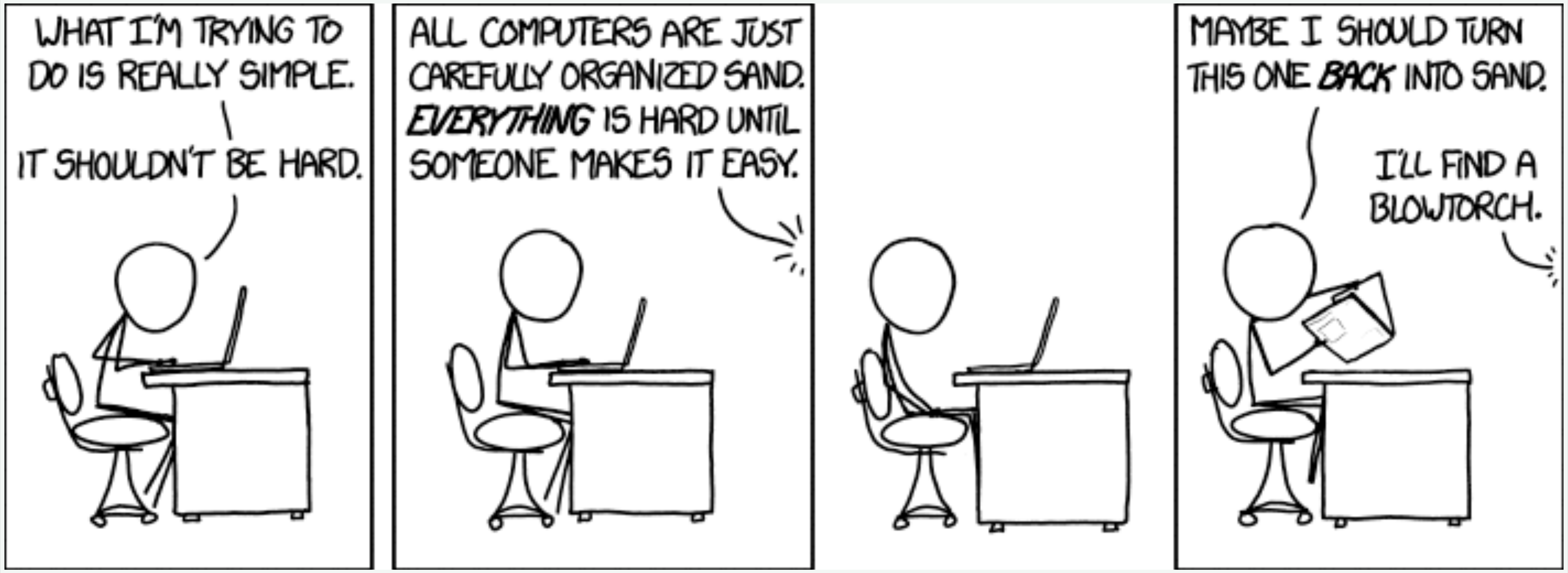
IT SHOULDN'T BE HARD...

FUTURE OF THE WEB


WHAT I'M TRYING TO DO IS REALLY SIMPLE.
IT SHOULDN'T BE HARD.
ALL COMPUTERS ARE JUST CAREFULLY ORGANIZED SAND. *EVERYTHING* IS HARD UNTIL SOMEONE MAKES IT EASY.
MAYBE I SHOULD TURN THIS ONE *BACK* INTO SAND.
I'LL FIND A BLOWTORCH.

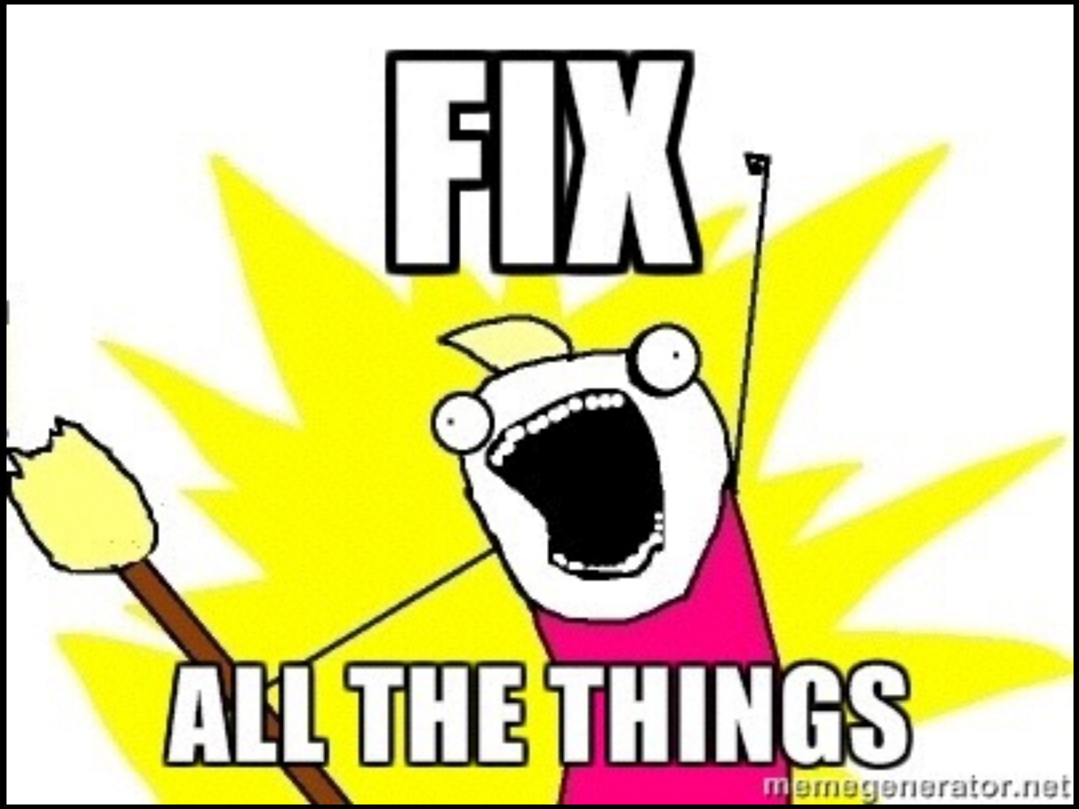
(six hours later) ARGH. How are these stupid microchips so durable?!
All I want is to undo a massive industrial process with household tools!

I'm sorry...

If you're only ever programmed for the web, I'm really really sorry …

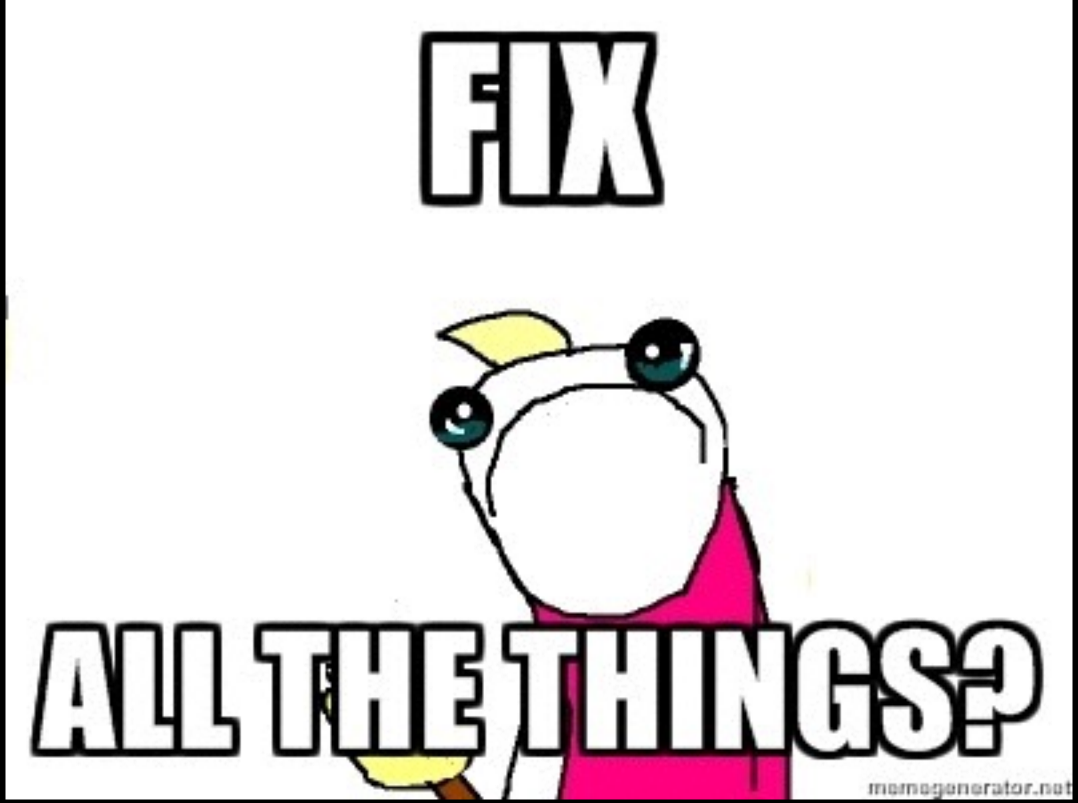
...because it's kind of broken.

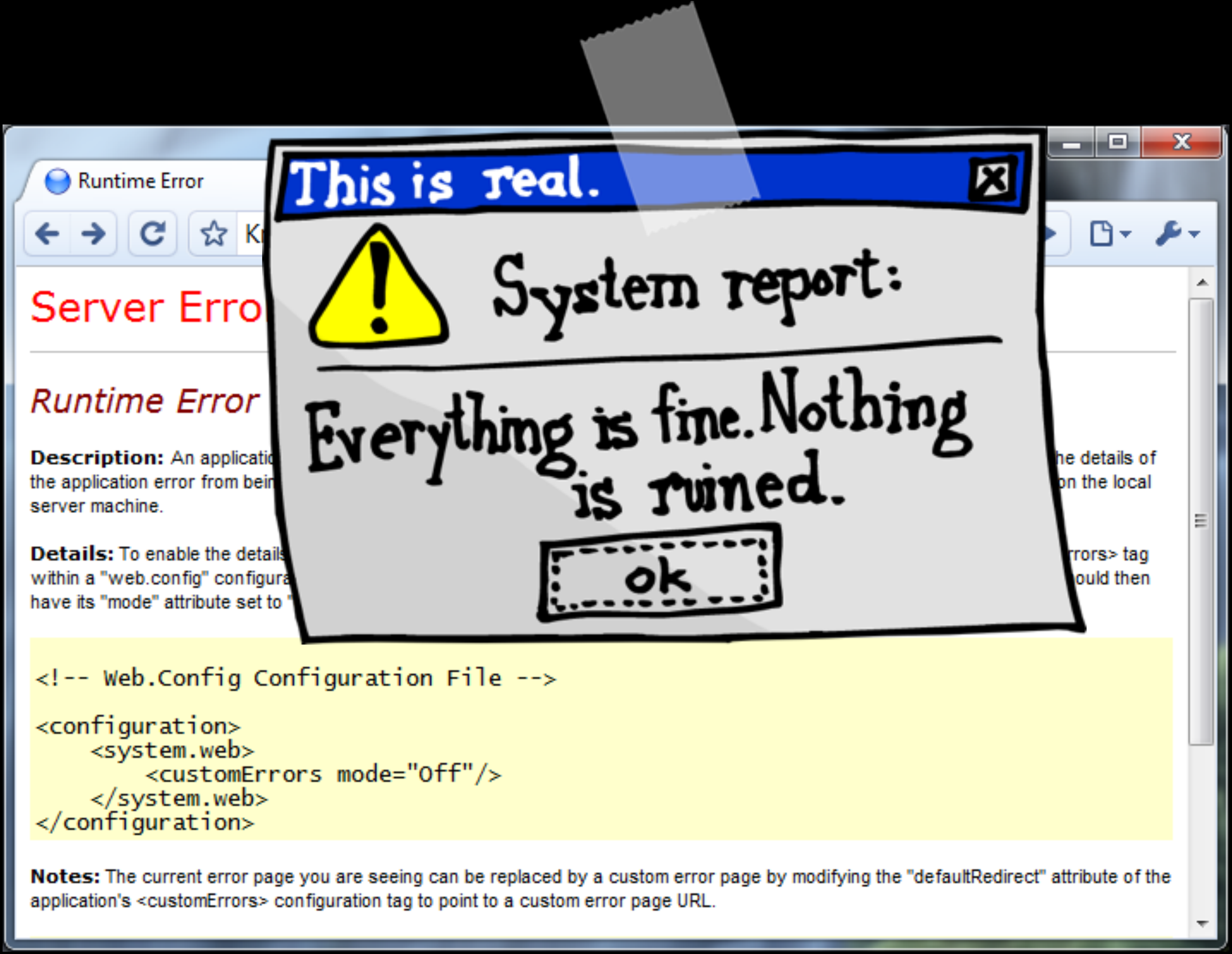
So let's fix the web!



“W3C can not remove any defects from the standards, they can only add more defects to it. That is what the web standards process does.”

- Douglas Crockford


FIX
ALL THE THINGS?
memegenerator.net



This is real.

System report:

Everything is fine. Nothing is ruined.

Server Erro

Runtime Error

Description: An applicati he details of the application error from bein on the local server machine.

Details: To enable the details rrors> tag within a "web.config" configura ould then have its "mode" attribute set to "

```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="Off"/>
    </system.web>
</configuration>
```

Notes: The current error page you are seeing can be replaced by a custom error page by modifying the "defaultRedirect" attribute of the application's <customErrors> configuration tag to point to a custom error page URL.

This should have been the title

WHY THE WEB IS TERRIBLE

- Insecure
- Complex

Let's break this web thing down.

And hope it works better than locked marketing gears.

WHY THE WEB IS TERRIBLE…

TRANSPORT

- HTTP - Insecure. Slow. Complex.
- DNS - Insecure. Vulnerable to confusion attacks.
- SSL - “And after almost 15 years we still don’t have it working”

I just want to send some cat pictures…

WHY THE WEB IS TERRIBLE…

LAYOUT

- HTML - Tags and trees. Hard to use. Human readable?
- CSS - “Crappy Style Sheets”
- DOM - “The worst API ever invented”

I just want this box on the left…
Maybe we can make it blue as a stretch goal.

WHY THE WEB IS TERRIBLE…

CODE

- JavaScript - De facto language? “A hot mess”
- Java, C#, Python, Ruby, Perl, PHP, etc etc
- C and C++ (emscripten/asm.js)

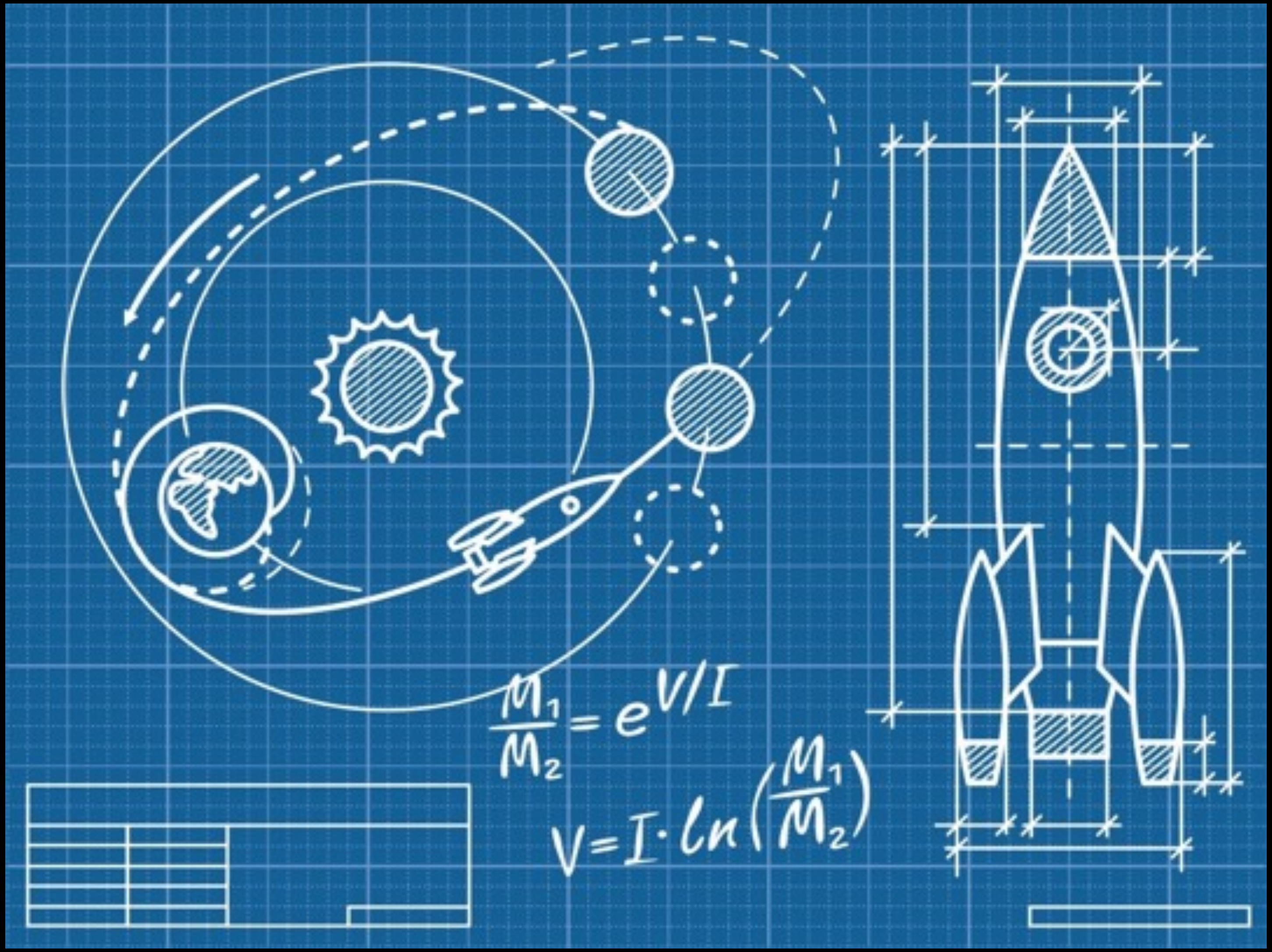
Standards? Screw it! I’ll make my own!

Fix the web?

SOUNDS LIKE A LOT OF WORK.

IS IT EVEN WORTH IT?

Web + "Business Rules" = ???


$\frac{M_1}{M_2} = e^{V/I}$
$V = I \cdot \ln\left(\frac{M_1}{M_2}\right)$

I think rocket science might be easier...

LET'S CHANGE THE TITLE AGAIN!

WHY THE WEB IS AWESOME

- Decentralized
- Ubiquitous
- Participatory
- Cat pictures!!!!

Cat pictures make everything better!

WHY THE WEB IS AWESOME!

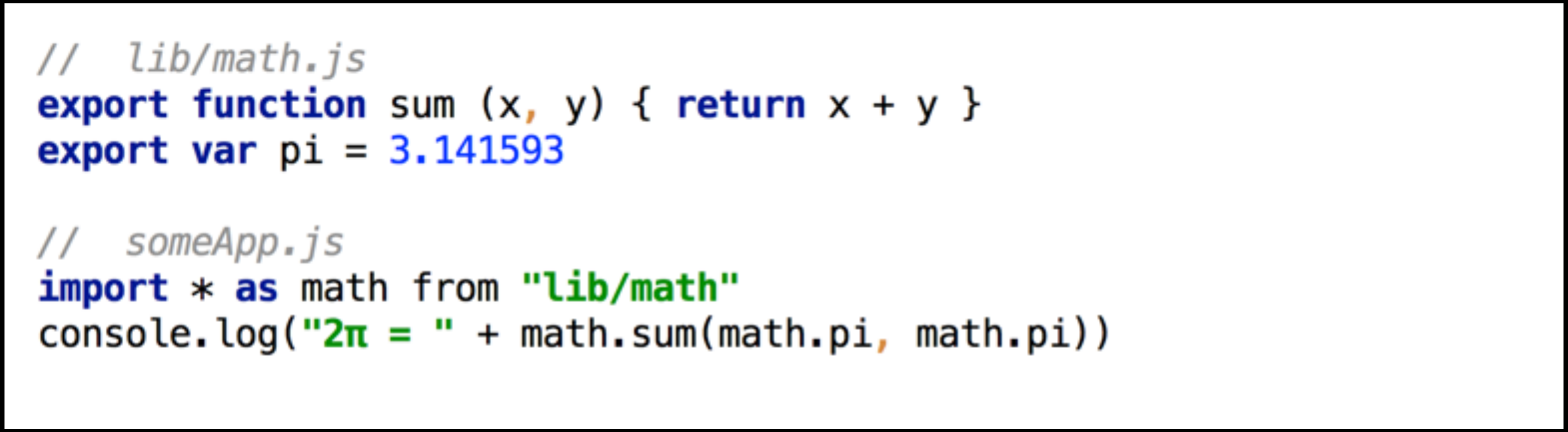
NEVER BEEN BETTER TO BE A DEVELOPER

- Evolving tools
- Evolving IDEs
- Evolving languages

What? WEB is evolving!

5 NEW JAVASCRIPT FEATURES

feature[0]: MODULES

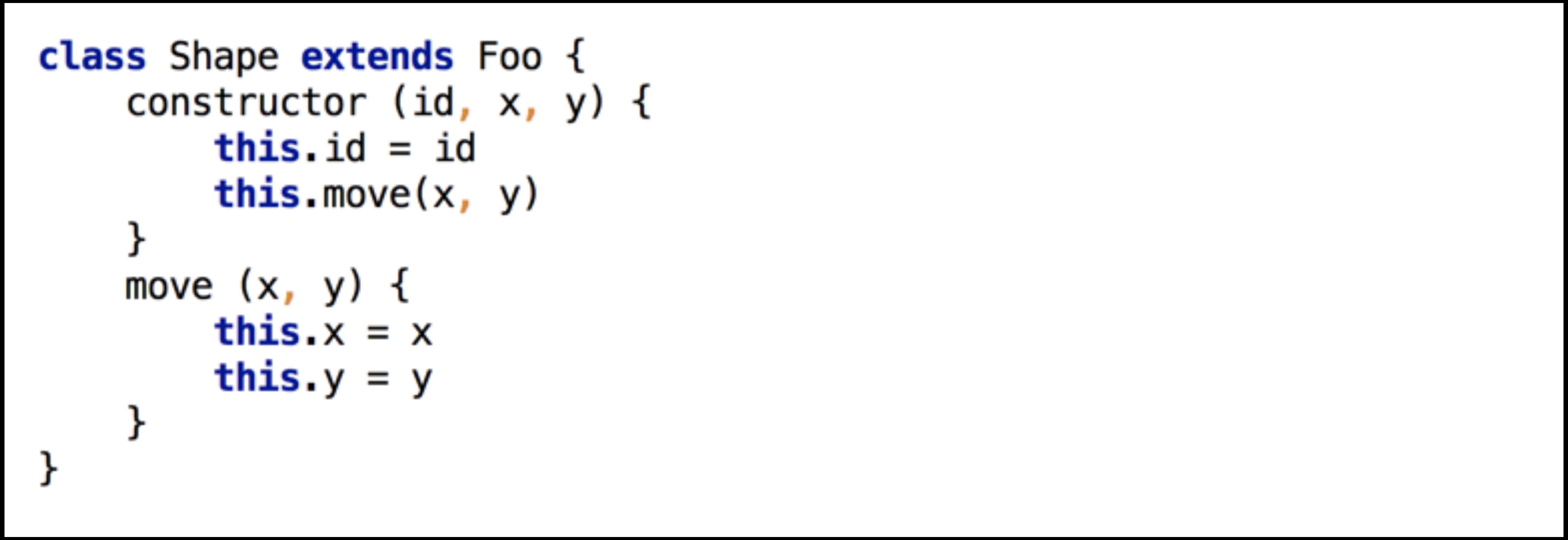


```
//  lib/math.js
export function sum (x, y) { return x + y }
export var pi = 3.141593

//  someApp.js
import * as math from "lib/math"
console.log("2π = " + math.sum(math.pi, math.pi))
```

Replaces RequireJS and other loaders

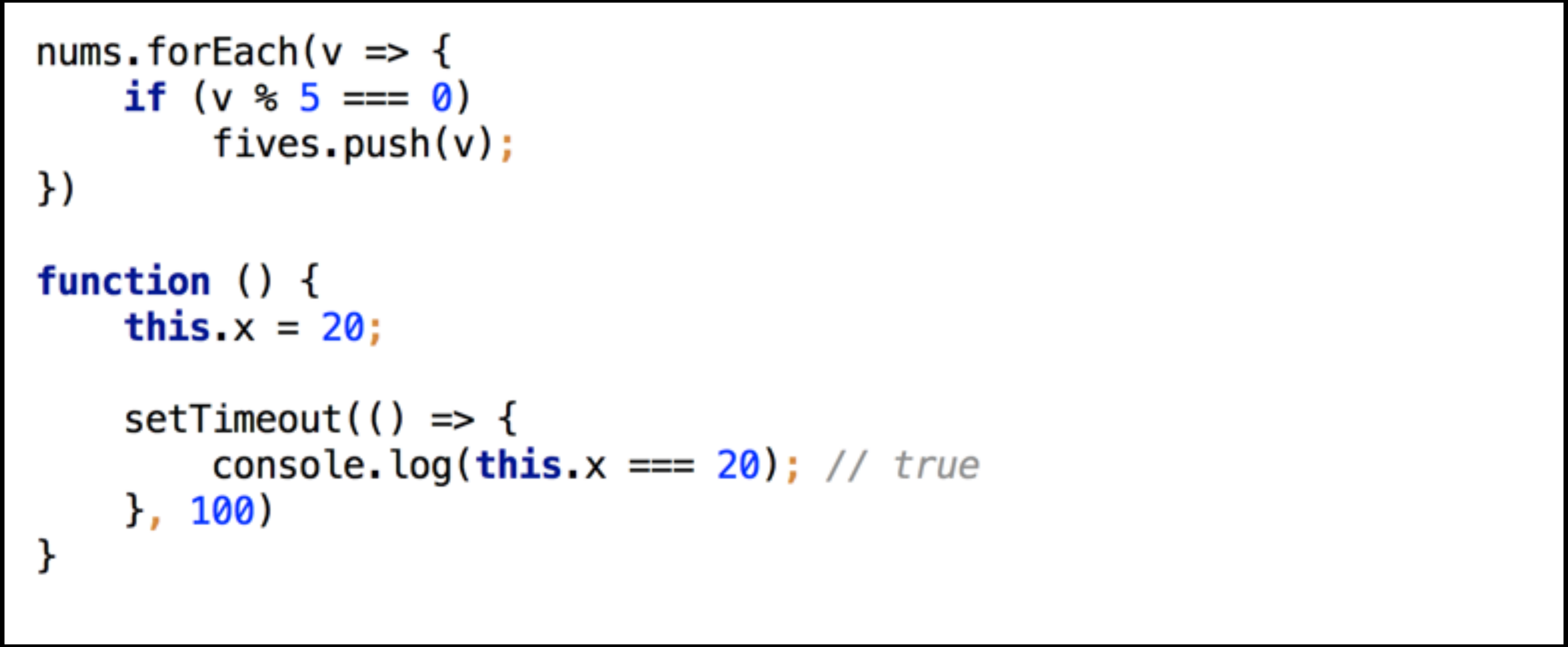
feature[1]: CLASSES



```
class Shape extends Foo {
    constructor (id, x, y) {
        this.id = id
        this.move(x, y)
    }
    move (x, y) {
        this.x = x
        this.y = y
    }
}
```

Replaces “Simple” Inheritance and JSDoc based typing

feature[2]: ARROW FUNCTIONS



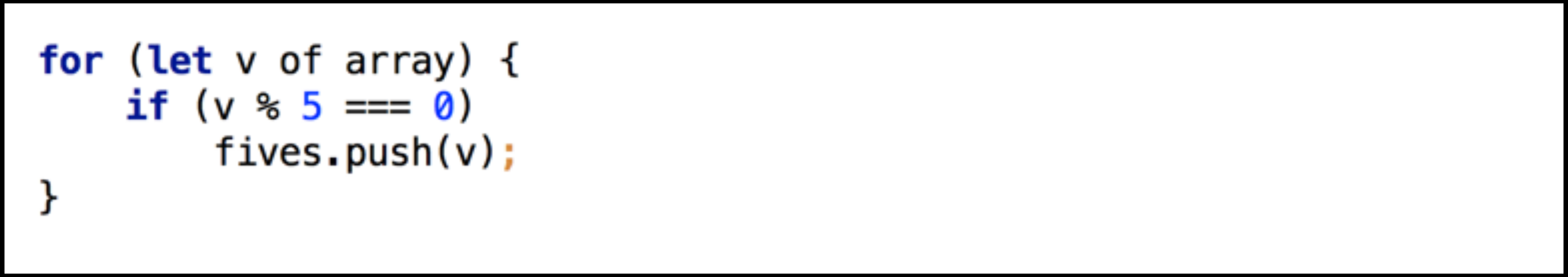
```
nums.forEach(v => {
    if (v % 5 === 0)
        fives.push(v);
})

function () {
    this.x = 20;

    setTimeout(() => {
        console.log(this.x === 20); // true
    }, 100)
}
```

Less typing and we don’t need extra vars. (var self = this;)

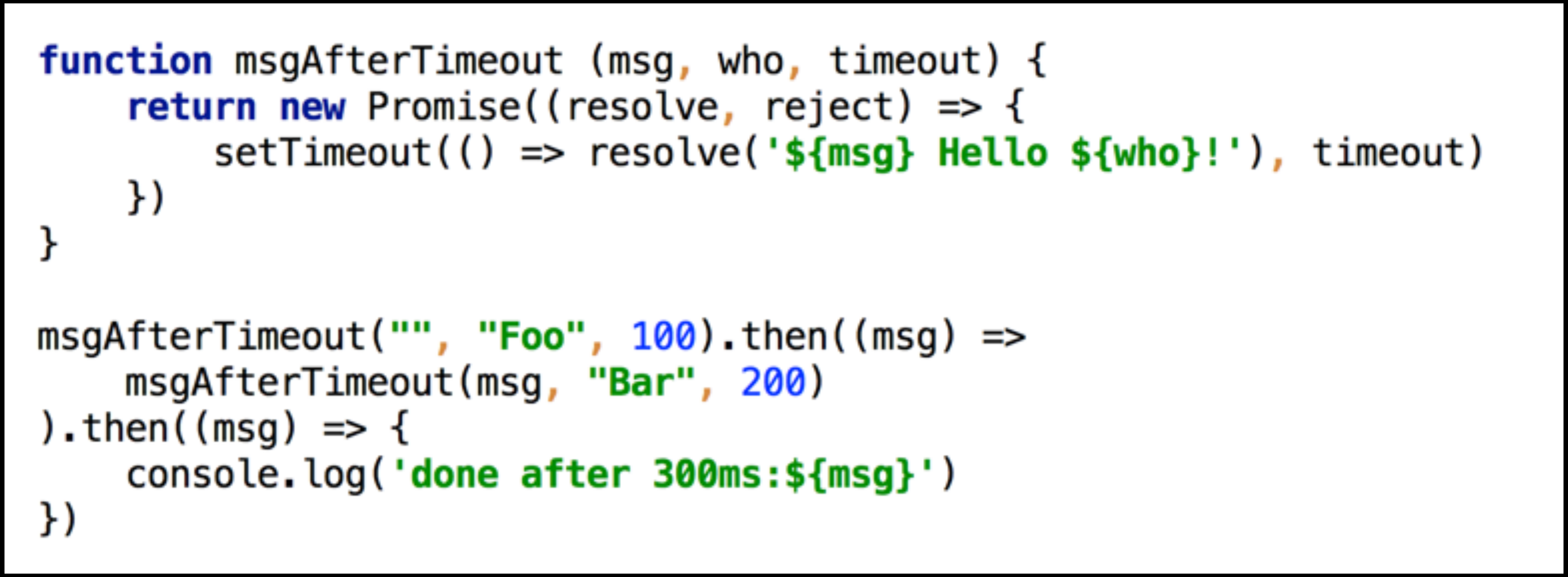
feature[3]: ITERATORS



```
for (let v of array) {
    if (v % 5 === 0)
        fives.push(v);
}
```

No more nested function inside a forEach!

feature[4]: PROMISES



```
function msgAfterTimeout (msg, who, timeout) {
    return new Promise((resolve, reject) => {
        setTimeout(() => resolve('${msg} Hello ${who}!'), timeout)
    })
}

msgAfterTimeout("", "Foo", 100).then((msg) =>
    msgAfterTimeout(msg, "Bar", 200)
).then((msg) => {
    console.log('done after 300ms:${msg}')
})
```

Bonus feature: string templates and string interpolation

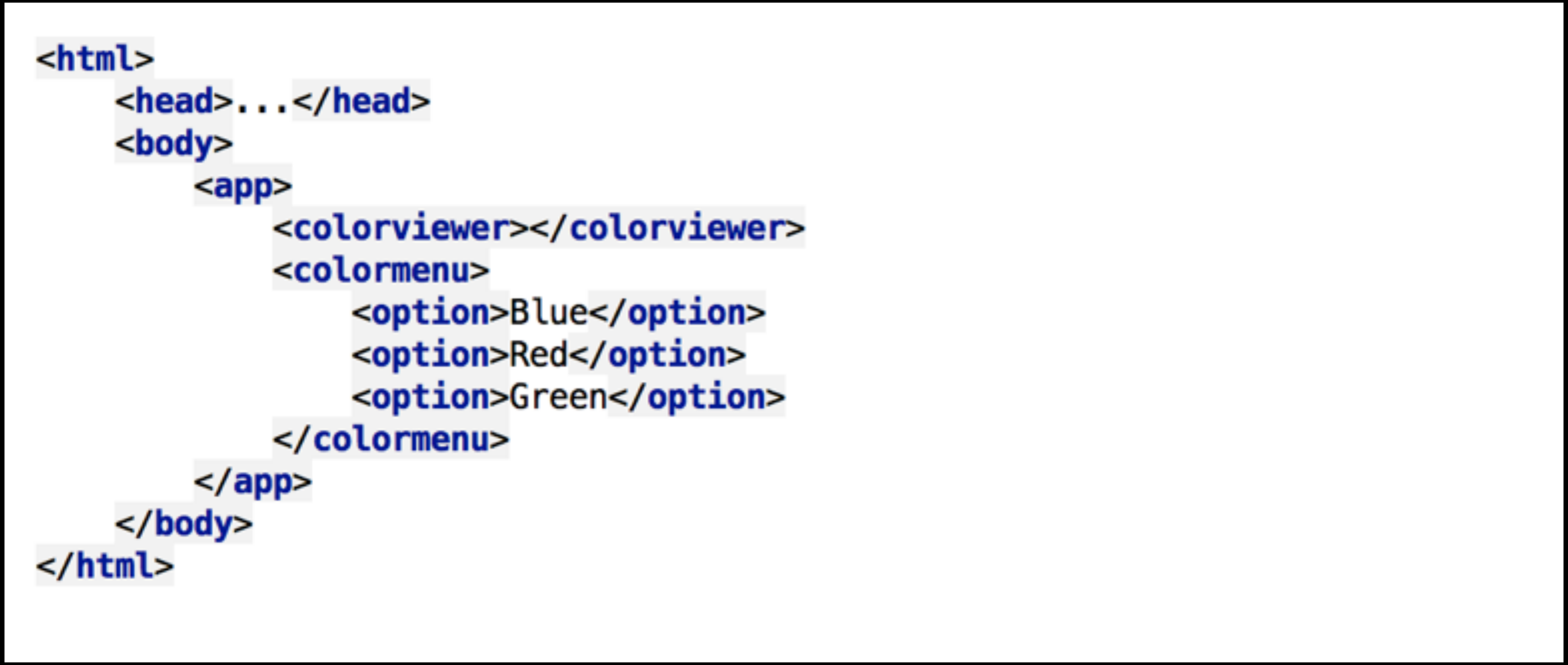
ES6 ALL THE THINGS!

- arrows
- classes
- enhanced object literals
- template strings
- destructuring
- default + rest + spread
- let + const
- iterators + for..of
- generators
- unicode
- modules
- module loaders
- map + set + weakmap + weakset
- proxies
- symbols

This is literally the logo…

BEYOND

WEB COMPONENTS

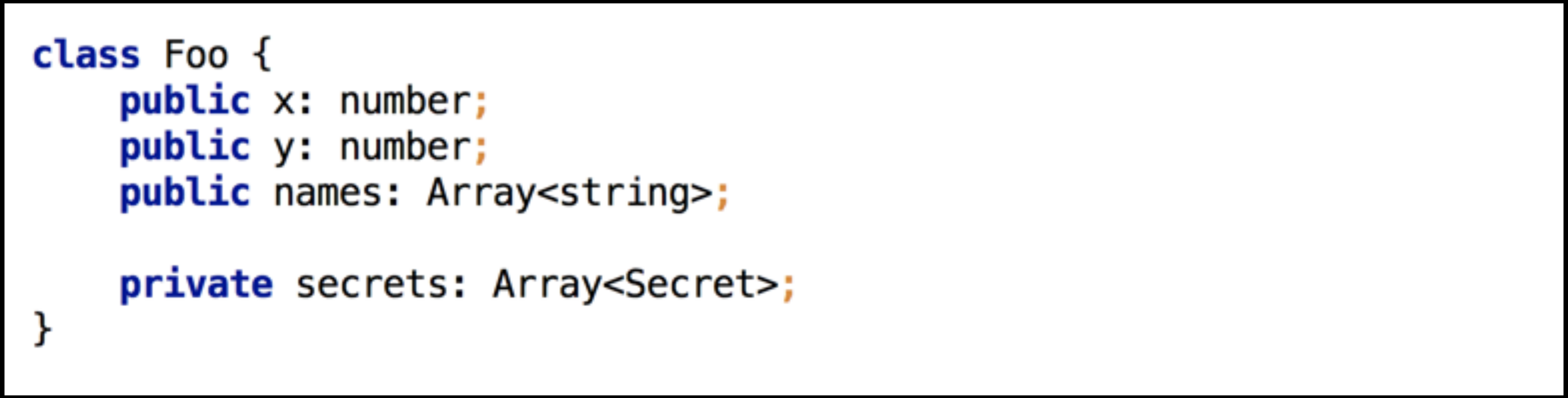


```
<html>
    <head>...</head>
    <body>
        <app>
            <colorviewer></colorviewer>
            <colormenu>
                <option>Blue</option>
                <option>Red</option>
                <option>Green</option>
            </colormenu>
        </app>
    </body>
</html>
```

This one is still pretty drafty (beyond ES6)

BEYOND

STRONG TYPES



```
class Foo {
    public x: number;
    public y: number;
    public names: Array<string>;

    private secrets: Array<Secret>;
}
```

Still in proposal for ESx, but with TypeScript…

DON'T WAIT FOR BROWSER TO CATCH UP

The Future is Today!

- ES6 - Available experimentally in Chrome
- TypeScript - ES6 superset. Transpires to ES5/ES6
- SystemJS - ES6 module polyfill. Supports AMD.
- Polymer - Framework independent components.

THE GIANTS ARE GETTING ALONG?

MicroSoft + Google = ???

HARMONY

- Participate
- Embrace change
- Never stop learning
- Experiment
- Play