

Berlin | November 20 - 21, 2018



# Seven things to make you a happier JavaScript developer

Christian Heilmann



# Seven ways to be a happier JavaScript developer



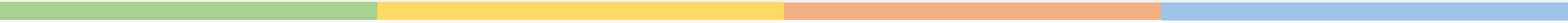
# 7 ways to be a happier JavaScript developer



- Understanding that JavaScript is not a language
- Concentrating on the now
- Limiting our development environment
- Making it harder to write bad code
- Getting to know our tooling
- Giving back to others
- Muffling the noise

1

Understanding that  
JavaScript is not a  
language...



# JavaScript is a lot of things...




- Far from perfect
- Everywhere
- Free to use
- Environment independent
- Freely documented
- Instantly gratifying

JavaScript's  
versatility means  
diverse needs.



- Web scripts
- WebView based solutions (Electron)
- Browser extensions
- Web based apps (PWA)
- Server-side solutions (Node)
- Convertible to binary formats (WebAssembly)
- Powering Robots (Nodebots)
- Packages (NPM)

JavaScript is much more now than we ever expected...



- A standardised language (TC39)
- An ecosystem
- A community
- An opportunity to do a lot with one language
- A chance to cause damage (performance, security)

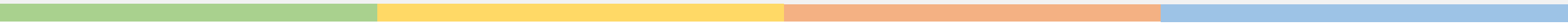
Relax - in a world  
of options nobody  
can be an expert  
in all of them.





# 2

Concentrating on  
the now...



Worries of the past  
shouldn't prevent  
us from creating...



Focusing exclusively  
on the next cool  
thing is as stifling  
and depressing



## Resources for developers, by developers.

🔍 Search the docs

[Web Technologies](#)



[Learn web development](#)



[Developer Tools](#)



**moz://a**

Proudly non-profit.  
Donate before Dec 31.

[Donate Now](#) →



### Learn the best of web development

Get the latest and greatest from MDN delivered straight to your inbox.

[Sign up now](#)



# # Fetch - LS

Global

78.35% + 0.06% = 78.41%

A modern replacement for XMLHttpRequest.

Current aligned Usage relative Date relative Show all

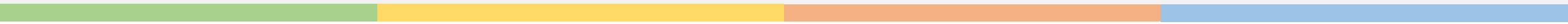
IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
		52	49						
		55	60		10.2				4
	15	56	61	10.1	10.3				5
11	16	57	62	11	11.1	all	62	11.4	6.2
	17	58	63	TP					
		59	64						
		60	65						

Notes Known issues (0) Resources (7) Feedback

- [Specification \[fetch.spec.whatwg.org\]](https://fetch.spec.whatwg.org) ref
- [Demo \[addyosmani.com\]](https://addyosmani.com) info
- [Polyfill \[github.com\]](https://github.com) info
- [IE platform status: Shipped \[status.modern.ie\]](https://status.modern.ie) official status
- [Chrome platform status: Enabled by default \[chromestatus.com\]](https://chromestatus.com) official status
- [Firefox platform status: 39 \[platform-status.mozilla.org\]](https://platform-status.mozilla.org) official status
- [WebKit platform status: Supported \[webkit.org\]](https://webkit.org) official status

3

Limiting our  
development  
environment





# The web development trinity

## Editor



- Where we code
- Where we tweak themes
- Where we mix spaces and tabs – *like animals*

## Terminal



- Where we do version control
- Where we run build tasks
- Where we deploy
- Where we tweak themes
- Where we assume everyone runs OSX

## Browser



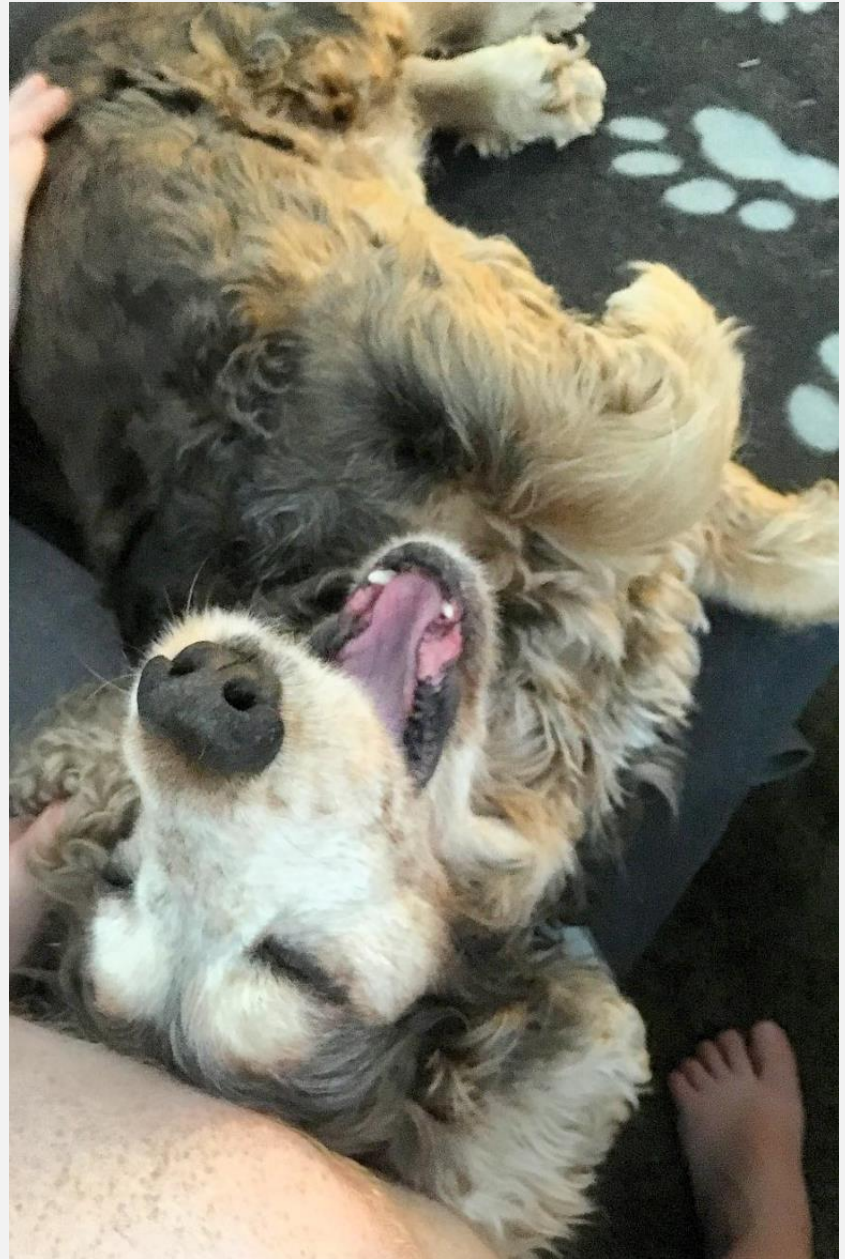
- Where we debug
- Where we check and tweak visual output
- Where we audit
- Where we annoy the end user with odd log messages.

That is a lot of  
duplication and  
multiplication  
with choice

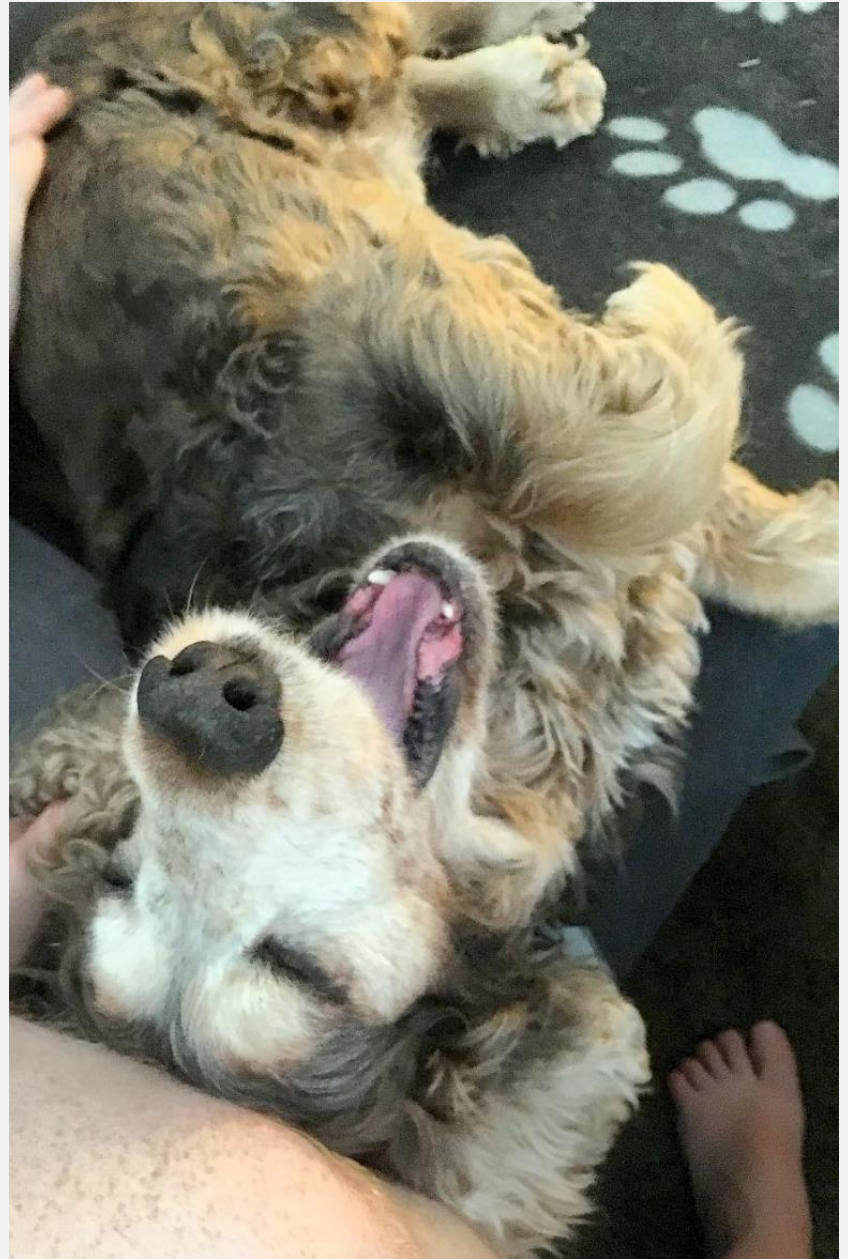




Tooling is starting to overtake that historically grown trinity.

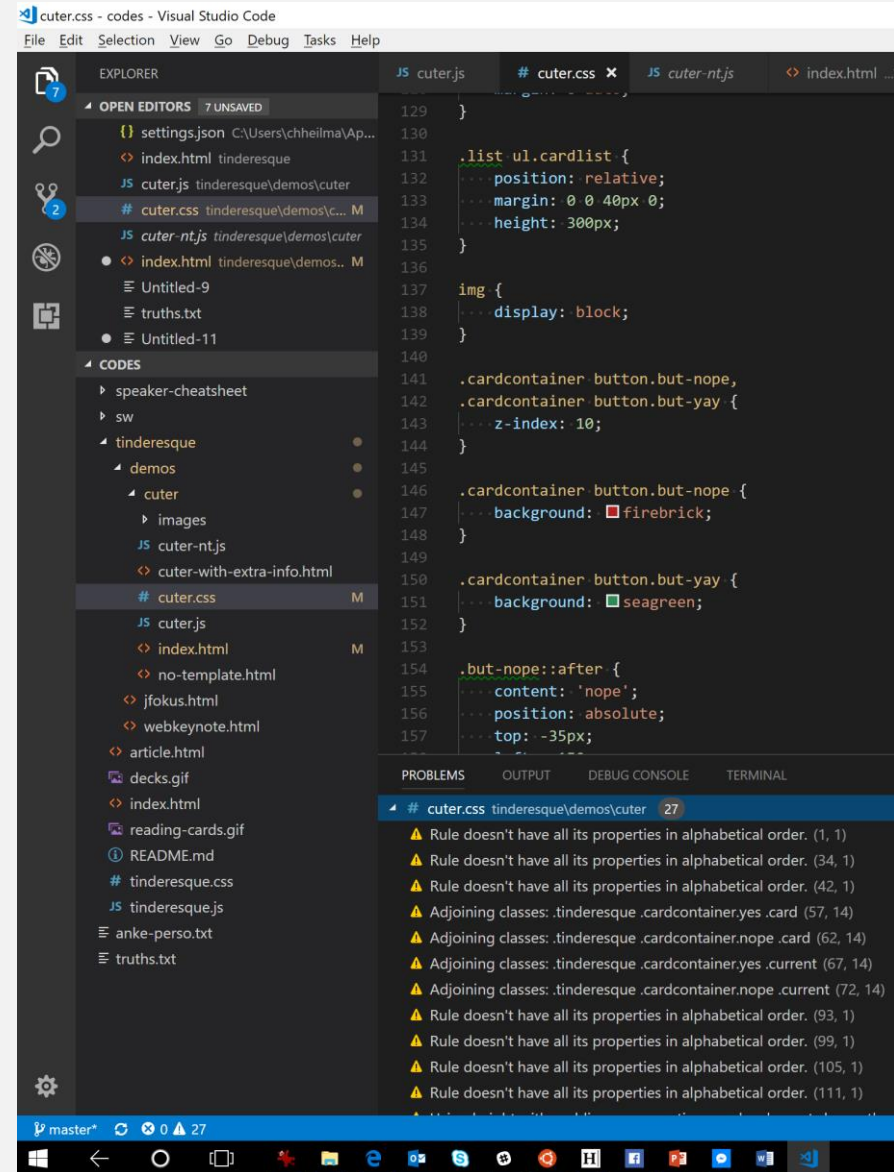


Good editors  
consolidate  
features to avoid  
context switching.



# VS Code (@code)

- Open Source, Cross-platform
- Hundreds of extensions
- Highly theme-able
- Built-in source control support
- Built-in terminal
- Integrates into build processes
- Written and extensible in TypeScript
- Straightens and lightens teeth with repeat use
- Possibly is good against hair loss
- Gluten free, macrobiotic

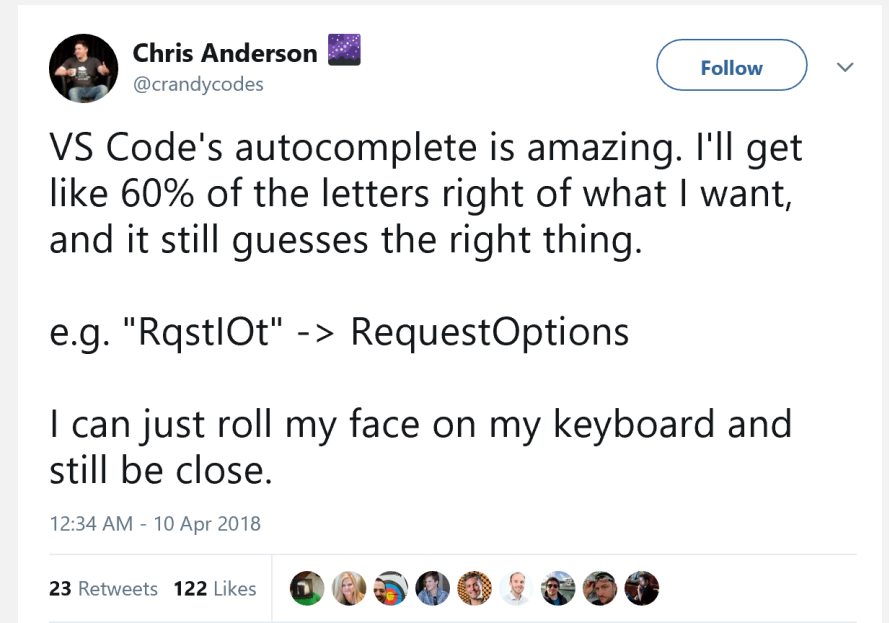


[code.visualstudio.com](https://code.visualstudio.com)



# VS Code (@code)

- Open Source, Cross-platform
- Hundreds of extensions
- Highly theme-able
- Built-in source control support
- Built-in terminal
- Integrates into build processes
- Written and extensible in TypeScript
- Straightens and lightens teeth with repeat use
- Possibly is good against hair loss
- Gluten free, macrobiotic



[twitter.com/crandycodes/status/983488436756627456](https://twitter.com/crandycodes/status/983488436756627456)

[code.visualstudio.com](https://code.visualstudio.com)

The image shows a screenshot of the Visual Studio Code editor interface. The main editor window displays the file `fibonacci.js` with the following code:

```
1
2 function fibonacci(num) {
3   if (num <= 1) {
4     return 1;
5   }
6   const result = fibonacci(num - 1) + fibonacci(num - 2);
7   console.log(`fib(${num}): ${result}`);
8   return result;
9 }
10
11 const fib = fibonacci(5);
12
```

The interface includes a sidebar on the left with the Explorer view showing the file structure, a search icon, and a terminal window at the bottom. The terminal window shows the prompt `sardinia:Auto Attach weinand$`. The status bar at the bottom indicates the current line and column (Ln 12, Col 1), tab size (4), encoding (UTF-8), line ending (LF), and language (JavaScript).

[code.visualstudio.com/updates/v1\\_22#\\_node-debugging](https://code.visualstudio.com/updates/v1_22#_node-debugging)



**About two years ago, I** begrudgingly opened Visual Studio Code (VS Code) for the first time. The only reason I even did so is that I was

working on a TypeScript project (also quite begrudgingly) and I was tired of fighting with the editor and the compiler and all of the settings that I needed to make a TypeScript project work.

Someone mentioned to me that TypeScript “just works” in VS Code and I was pleasantly surprised to find that they were right.

I expected that though. VS Code is from Microsoft, so it’s entirely expected that TypeScript would just work. What I did not expect was that virtually anything that I needed to do in VS Code “just worked” too. And I do mean *anything*. While this article focuses primarily on features for the web as well as JavaScript developers, VS Code has integrations with Go, Python, Java, and Ruby — just to name a few.

Over the next few months, I began to discover things in VS Code that I found remarkable. I made the official switch to VS Code, and I haven’t looked back since.



#### ABOUT THE AUTHOR

Burke Holland is a front-end developer living in Nashville, TN; the greatest city in the world. He enjoys JavaScript a lot because it’s the only way he ... [More about Burke...](#)

JANUARY 9, 2018 • [17 COMMENTS](#)

## Visual Studio Code Can Do That?

# [JavaScript](#) <sup>177</sup>

#### TABLE OF CONTENTS

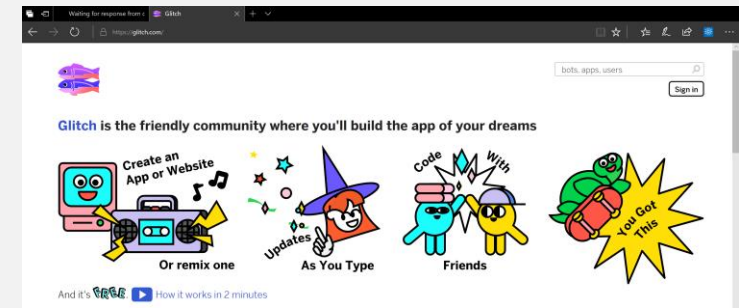


- 01 [Interactive Playground](#)
- 02 [Font Ligatures](#)
- 03 [Keyboard Mappings](#)
- 04 [Settings Sync](#)
- 05 [Emmet](#)
- 06 [Intellisense In JSON Files](#)

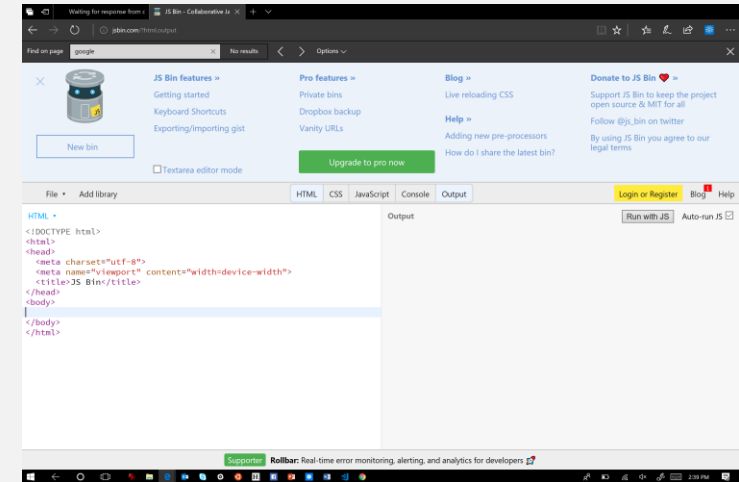
Instead of using  
your own  
machine, why not  
try online before?



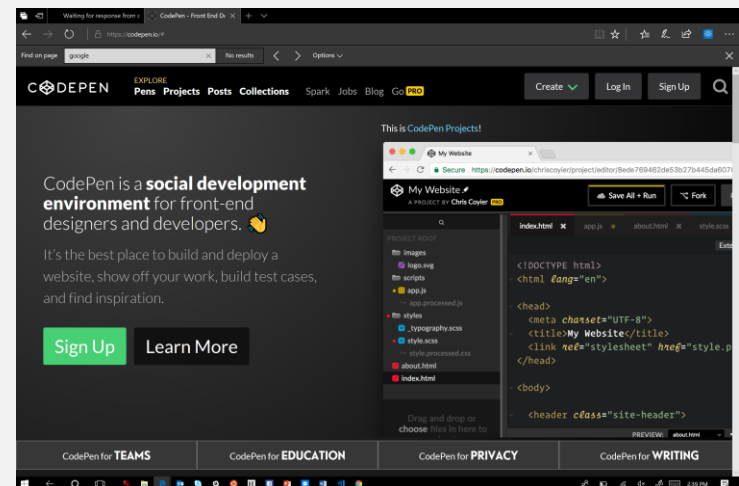
glitch.com



jsbin.com



codepen.io



# 4 Making it harder to write bad code.

## Introduce first topic/question

1. When we talk about enterprise aplications and the web, what's are the pain points?
2. What can be done about that? How could companies create happier users



Applying a  
stricter ruleset  
pre-empts  
mistakes

---

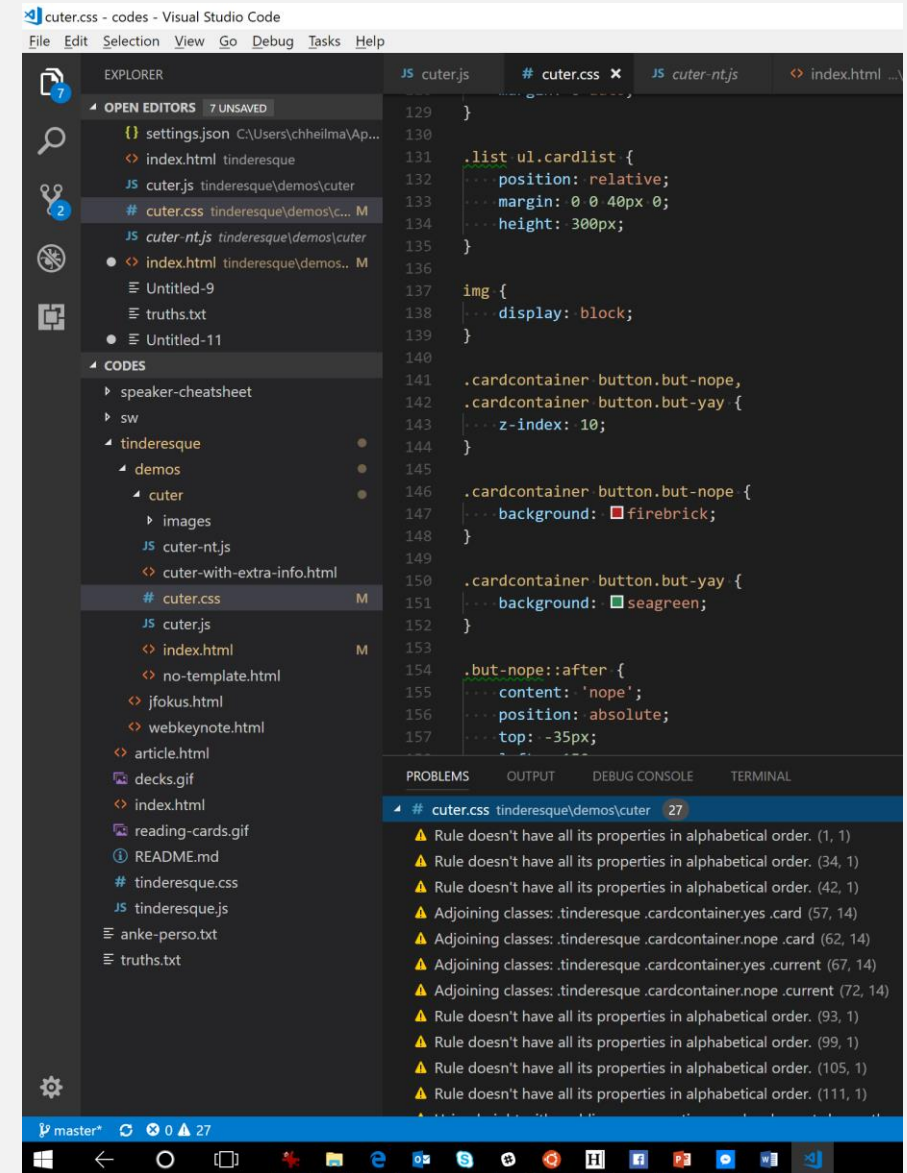


```
index.js — vs-code-can-do-that
index.js x
1  rick = () => {
2      var messages = [
3          'Never gonna give you up',
4          'Never gonna let you down',
5          'Never gonna run around and desert you'
6      ];
7
8      for (i = 0; i <= messages.length; i++) {
9          console.log(messages[i]);
10     }
11 };
12
13 rick();
14
```

0 0 Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript ESLint! Prettier: ✓

# Linting > Debugging

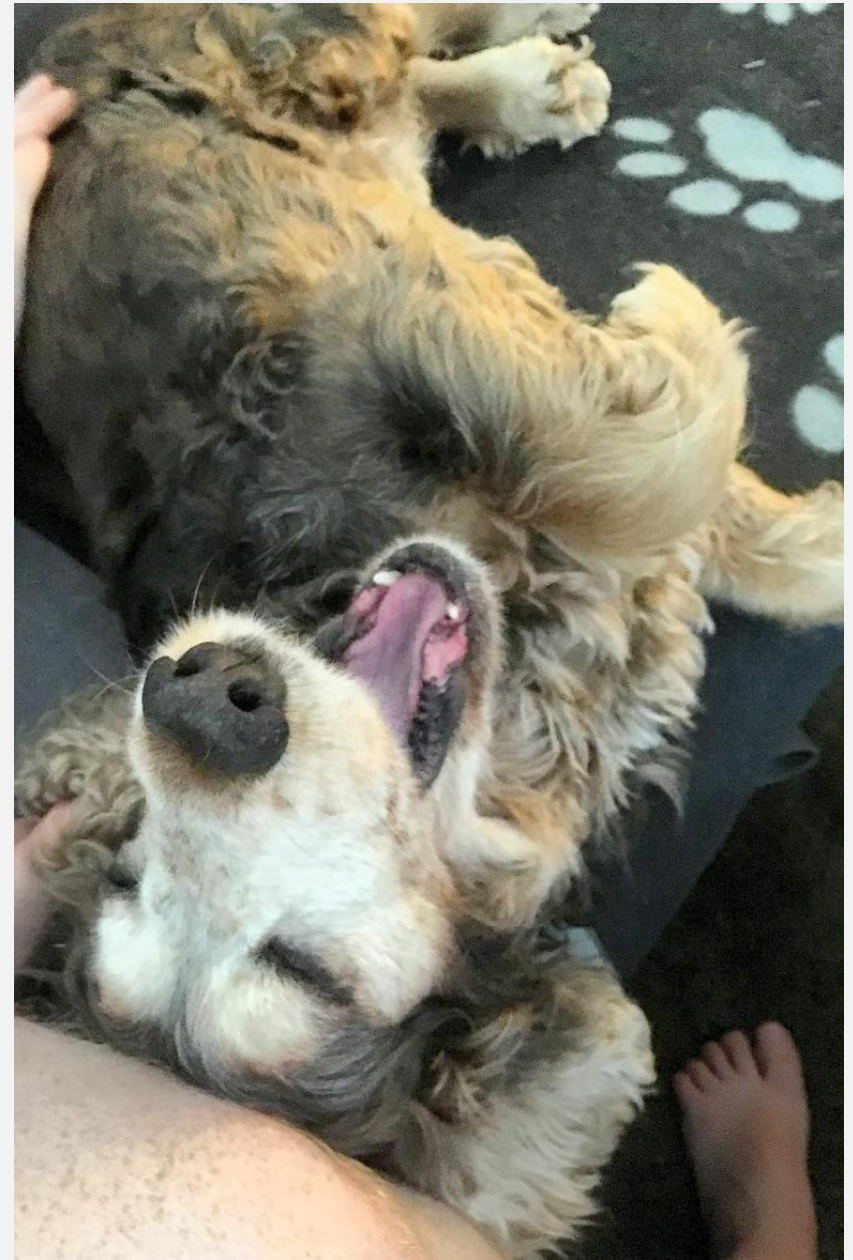
- Find mistakes while you make them
- Based on the experience and consensus of many others
- Learn from explanations of linting results
- Install and configure, or use in-built linting.



The screenshot shows the Visual Studio Code interface with a CSS file named 'cuter.css' open. The Explorer sidebar on the left shows the project structure, including a 'demos' folder containing 'cuter' and 'cuter-nt.js'. The main editor area displays the CSS code with several linting errors highlighted in yellow. The Problems panel at the bottom right lists these errors, such as 'Rule doesn't have all its properties in alphabetical order' and 'Adjoining classes'. The code in the editor includes rules for a list, an image, and buttons, with some properties like 'background' and 'position' being highlighted in red and green respectively.

Moving from  
knowing .\*  
to  
learning by  
making mistakes

---



Adding custom  
linting and  
validation to your  
dev and release  
process.



JS A JS Foundation Project

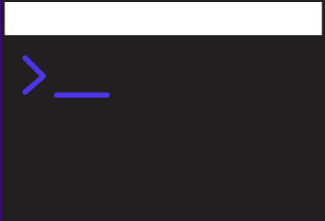
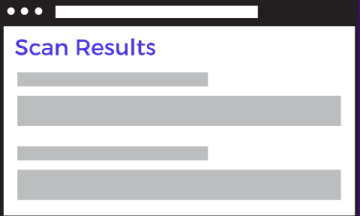
webhint

Use webhint to improve your website

webhint is a linting tool that will help you with your site's accessibility, speed, security and more, by checking your code for best practices and common errors. Use the online scanner or the CLI to start checking your site for errors.

TRY THE ONLINE SCANNER

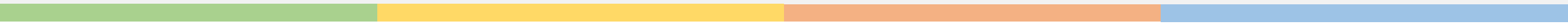
GET STARTED WITH THE CLI



webhint.io

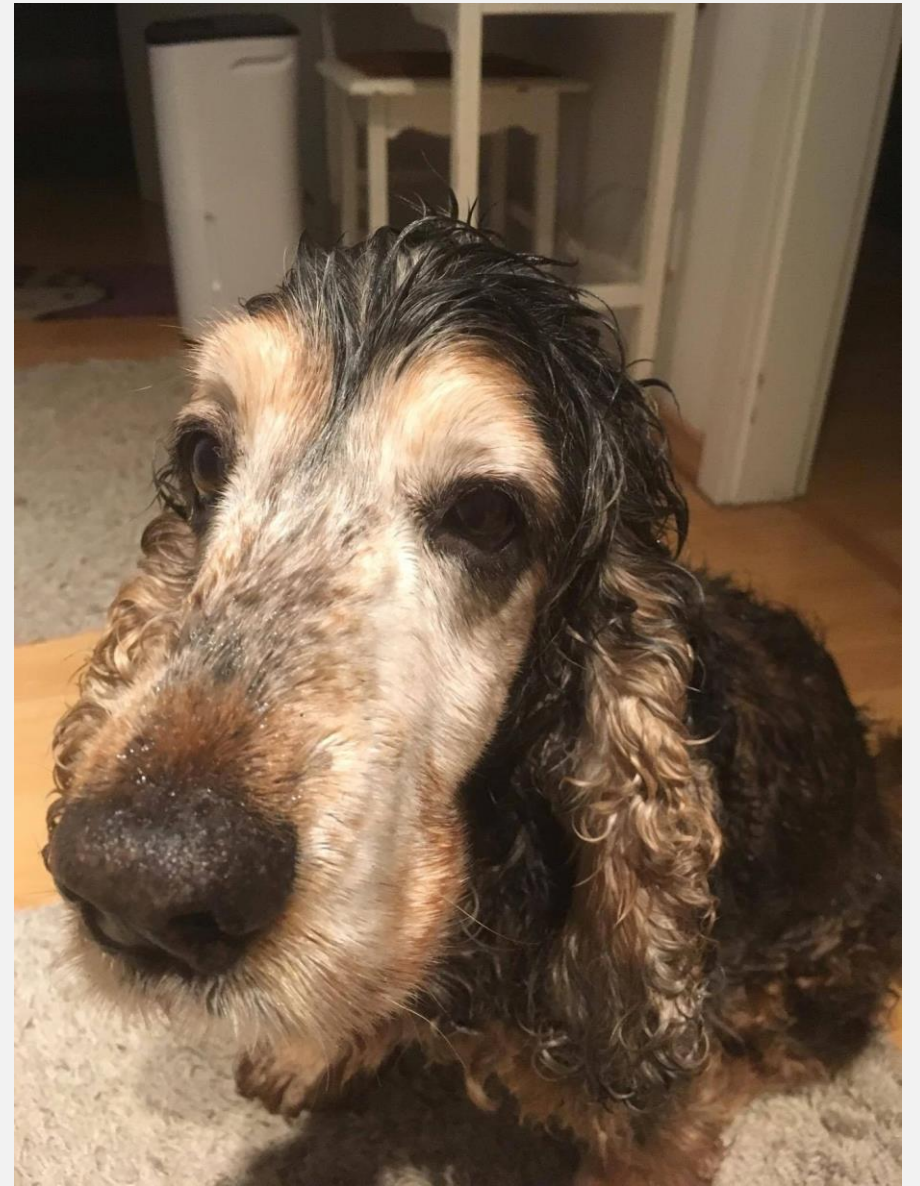
5

Getting to know our  
tooling

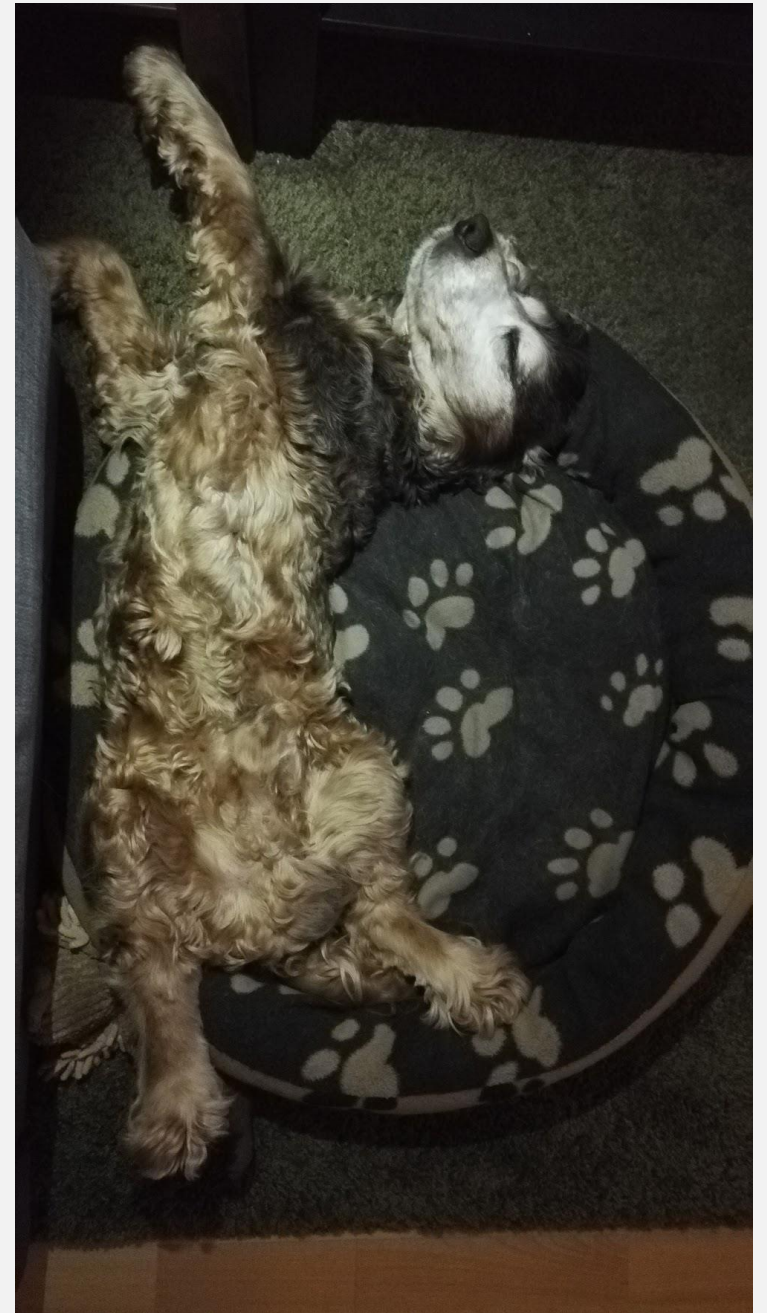




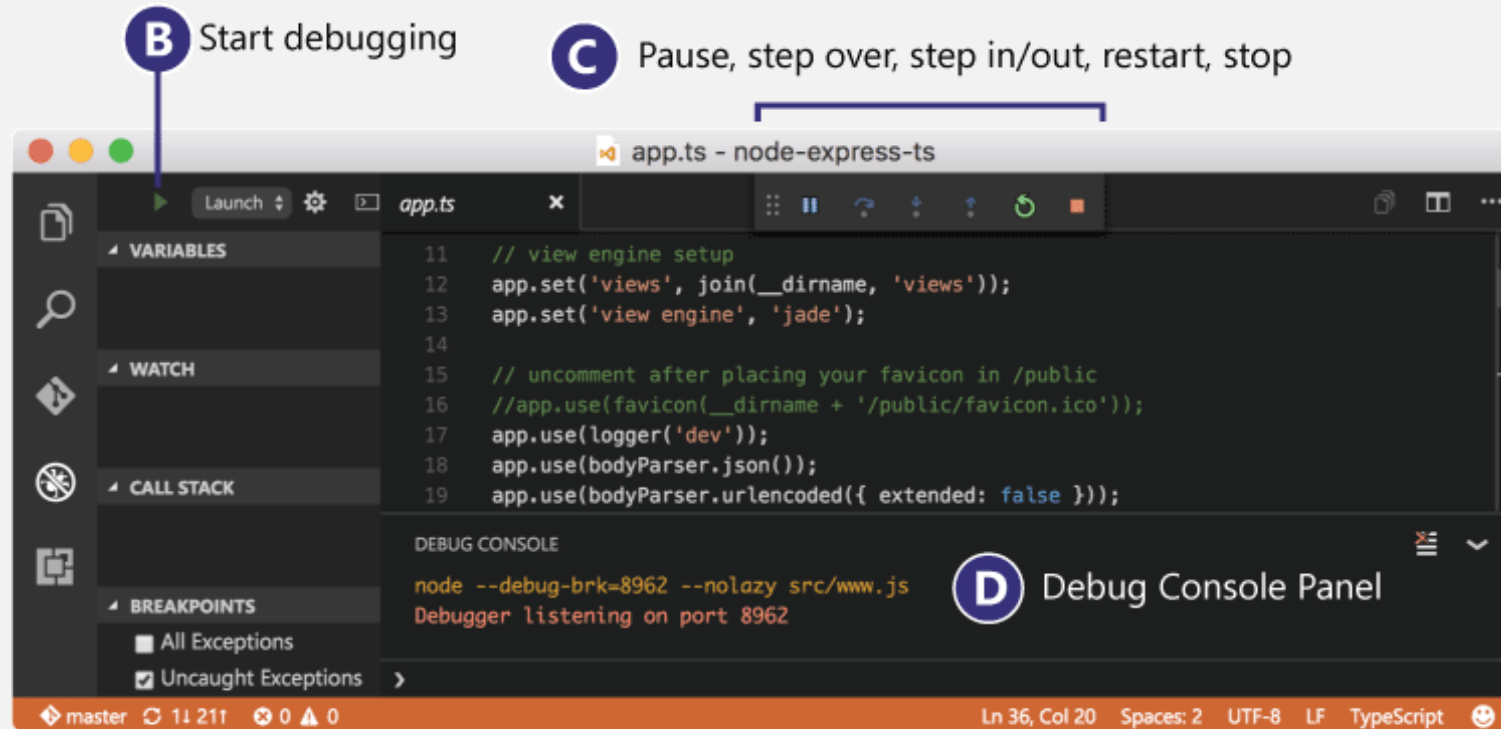
It is true that our  
tools have become  
much more complex.



Knowing them gives  
you superpowers  
and insights how to  
build great solutions



Step one is to stop using *console.log()* and use breakpoints instead.



A Debug Side Bar

Panel

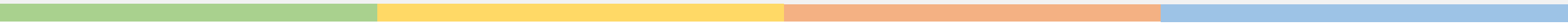
[code.visualstudio.com/Docs/editor/debugging](https://code.visualstudio.com/Docs/editor/debugging)

[smashingmagazine.com/2018/02/javascript-firefox-debugger/](https://smashingmagazine.com/2018/02/javascript-firefox-debugger/)



6

Giving back to  
others



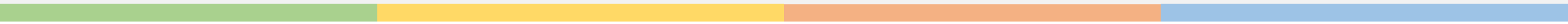
# Take part in the JavaScript community...



- Help document
- Help clean up and send pull requests
- Publish your own work as open source
- Enjoy meetups and events
- Help by example, not by telling people what they should do

# 7

Muffling the noise...



Not everybody  
cares about the nuts  
and bolts and has  
endless time to  
invest...



head  
@headjs\_ru

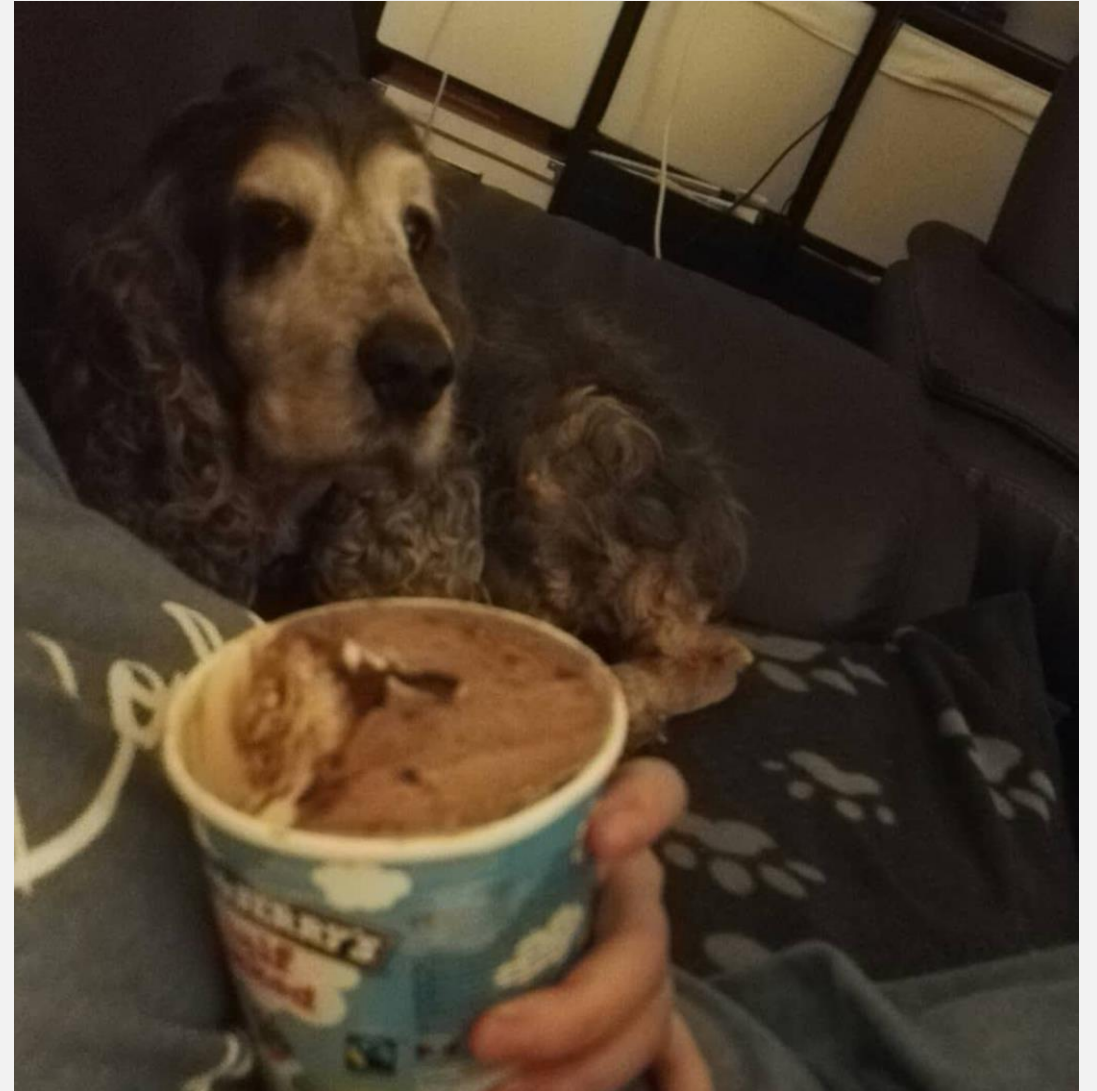
Follow




6:56 AM - 22 Jan 2018

[twitter.com/headjs\\_ru/status/955333399564161024](https://twitter.com/headjs_ru/status/955333399564161024)

It is easy to get  
tempted to sell  
people your  
experience as  
“best practice”




Find the  
largest item in  
the array...



```
let arr = [20, 10, 5, 10];
```

Once learned,  
never  
forgotten...



```
let arr = [20, 10, 5, 10];
```

```
const largestInArray = function(arr) {  
  let temp = 0;  
  for (let i = 0; i < arr.length; i++) {  
    if (array[i] > temp) {  
      temp = array[i];  
    }  
  }  
  return temp;  
}
```

# Loop optimisation




```
let arr = [20, 10, 5, 10];
```

```
const largestInArray = function(array) {  
  let temp = 0;  
  let all = array.length;  
  while (all-->0) {  
    if (array[all] > temp) {  
      temp = array[all];  
    }  
  }  
  return temp;  
}
```



Using native  
methods  
instead of "if"...



```
let arr = [20, 10, 5, 10];
```

```
const largestInArray = function(array) {  
  let temp = 0;  
  let all = array.length;  
  while (all-->0) {  
    temp = Math.max(array[all], temp);  
  }  
  return temp;  
}
```


Looking up  
array  
methods..



```
let arr = [20, 10, 5, 10];
```

```
const largest = arr.sort().reverse()[0];
```

Embracing  
new language  
features...



```
let arr = [20, 10, 5, 10];
```

```
const arrMax = arr => Math.max(...arr);
```

Which one is  
the best?



# It depends™



- Who is writing the code?
- Where does it run?
- How clean is the data we deal with?
- What horrible old environment needs support?
- Who will maintain it?
- How likely is it ever to change?
- How will it be used?

# Validate and triage




- Let's not feel bad if not everything in the JS world excites us
- Not everything hot and cool is ready to replace what we are comfortable with.
- If it doesn't make our lives easier, it is OK to not use it – hype moves fast and forgets fast.
- Let's be excited, but also ready to give it a "meh"

# Projection isn't good...



- What makes us effective can be very subjective
- It makes sense to talk about our success.
- It makes less sense trying to force others to do the same
- New approaches lead to new happy paths, let's not discourage people from finding theirs.

Let's not kid ourselves about the bleeding edge...



- The amazing new idea of today very often becomes the problem of tomorrow
- Production code moves less fast than we think
- Our end users aren't guinea pigs or canaries
- It is the bleeding edge, make sure you have enough blood to give



Let's make this a  
great community...



- Let's be the people we'd like to have met when we started
- Let's be kind
- Let's be supportive
- Let's allow people to learn by making mistakes and letting off steam
- Let's not get lost in pointless drama

# Thanks!

Chris Heilmann  
@codepo8



Terminal photo by [Terminal 5 Insider](#)  
Editor photo by [John Oxley Library](#)  
Browser photo by [Joseph Brent](#)

