Seven ways to be a happier JavaScript developer



Chris Heilmann (@codepo8) April 2018

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

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

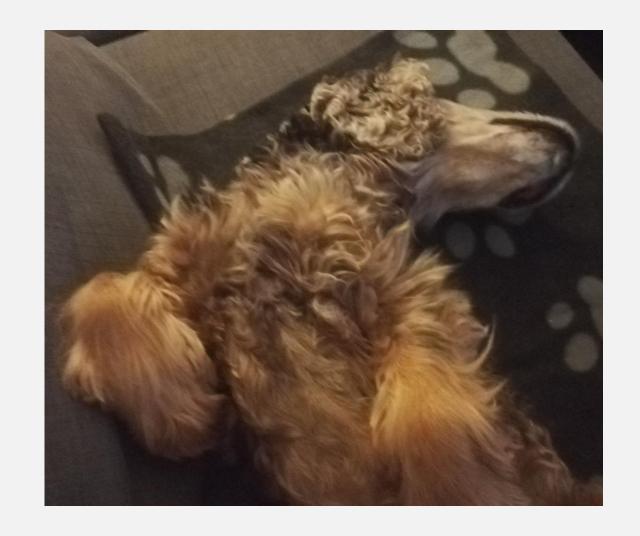
JavaScript's versatility means diverse needs.

- 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...

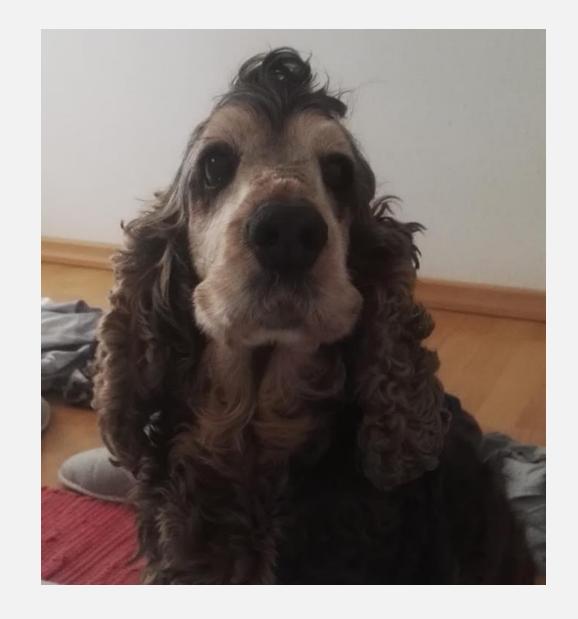
- A standardised language (TC39)
- An ecosystem
- A community
- An opportunity to do a lot with one language
- A chance to cause damage (performance, security)

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



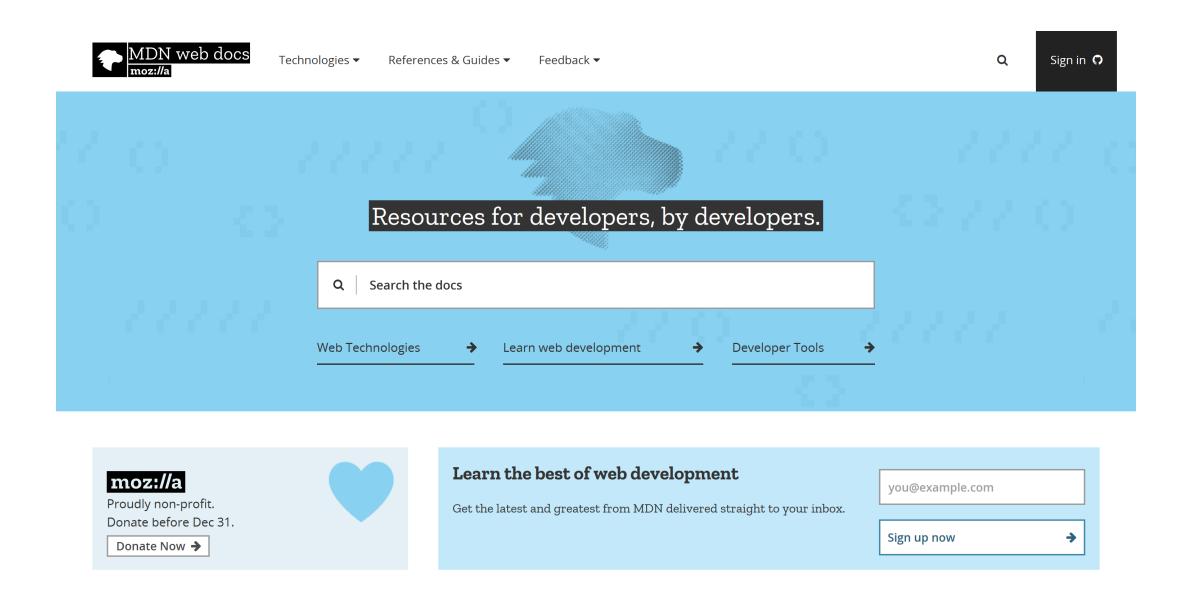
Concentrating on the now...

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



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







caniuse.com

Limiting our development environment

The web development trinity

Editor

Terminal

Browser



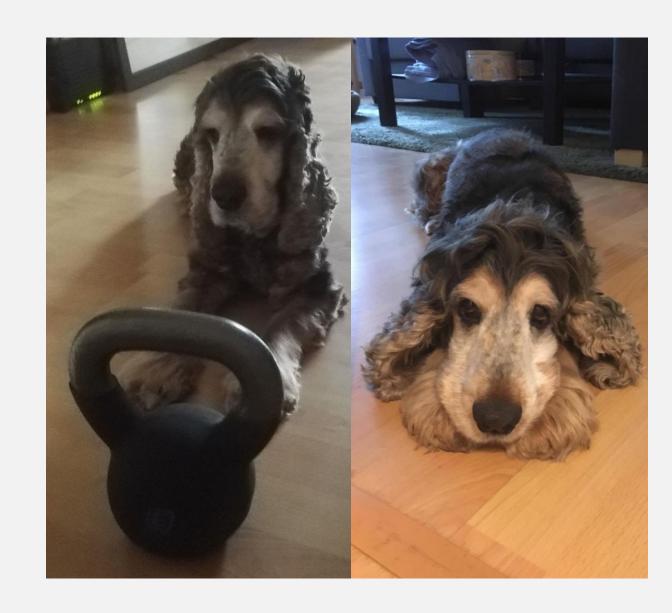




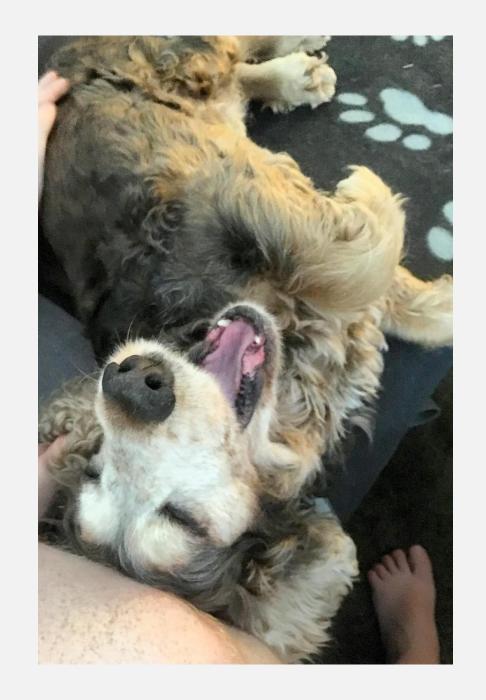
- Where we code
- Where we tweak themes
- Where we mix spaces and tabs – like animals
- Where we do version control
- Where we run build tasks
- Where we deploy
- Where we tweak themes
- Where we assume everyone runs OSX

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

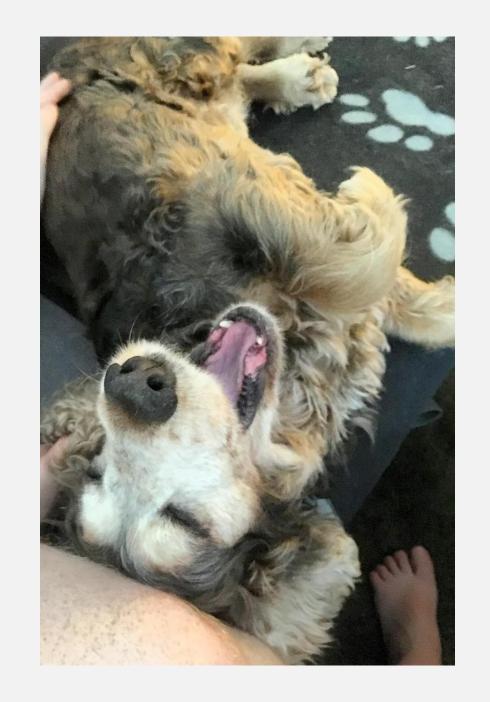
That is a lot of duplication and multiplication with choice



Tooling is starting to overtake that historically grown trinity.

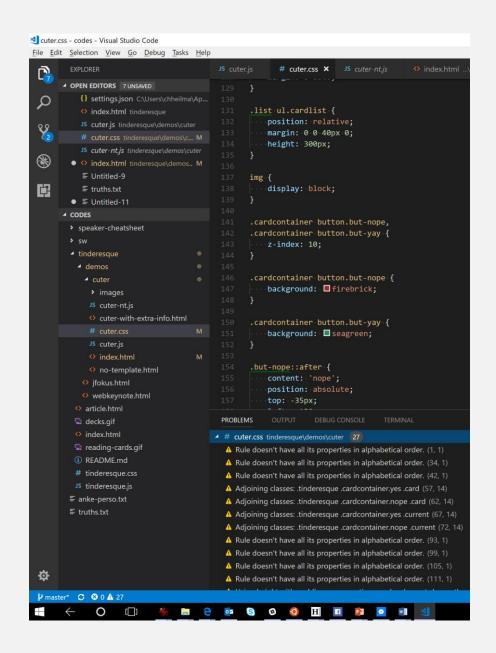


Good editors consolidate features to avoid context switching.



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

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

code.visualstudio.com

```
fibonacci.js - Auto Attach
                                 JS fibonacci.js
                                                                                                                  Ⅲ ×
        EXPLORER
      ▲ OPEN EDITORS
                                  2 	☐ function fibonacci(num) {
          JS hello world.js
                                           if (num ≤ 1) {
          JS fibonacci.js
                                               return 1;
      ▲ AUTO ATTACH
        > .vscode
                                           const result = fibonacci(num - 1) + fibonacci(num - 2);
       JS fibonacci.js
                                           console.log(`fib(${num}): ${result}`);
       Js hello world.js
                                           return result;
中
                                       const fib = fibonacci(5);
                                                TERMINAL
                                  PROBLEMS
                                                                                1: bash
                                sardinia:Auto Attach weinand$
                                             B
Ø 0 A 0
                                                                          Ln 12, Col 1 Tab Size: 4 UTF-8 LF JavaScript
```

code.visualstudio.com/updates/v1_22#_node-debugging



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..



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.

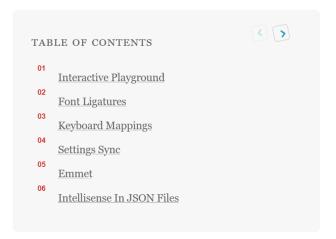
I expected that though. VS Code is from Microsoft, so it's entirely expected that TypeScript would just work. What I did not expect was that virtually anything that I needed to do in VS Code "just worked" too. And I do mean anything. While this article focuses primarily on features for the web as well 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.

JANUARY 9, 2018 • 17 COMMENTS

Visual Studio Code Can Do That?

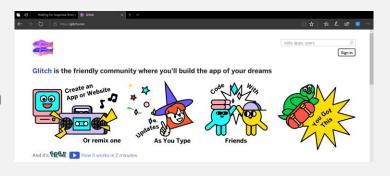
JavaScript 177



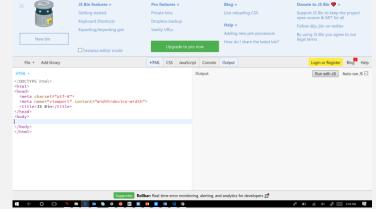
smashingmagazine.com/2018/01/visual-studio-code vscodecandothat.com

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

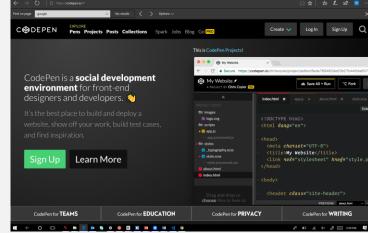
glitch.com



jsbin.com



codepen.io



Making it harder to write bad code.

Introduce first topic/question

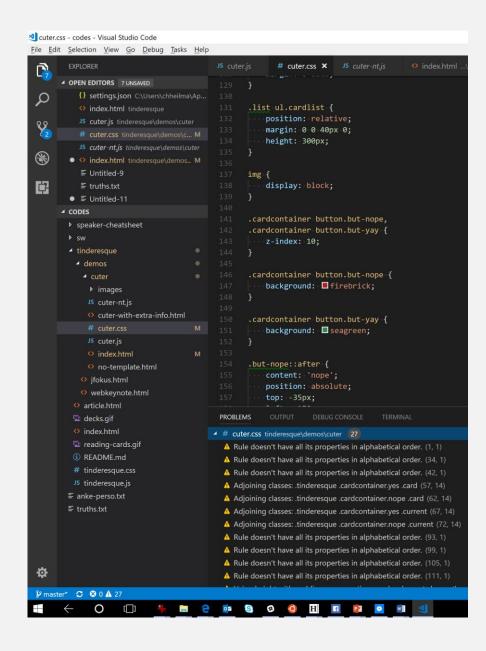
- 1. When we talk about enterprise <u>aplications</u> and the web, what's are the pain points?
- 2. What can be done about that? How could companies create happier users

Applying a stricter ruleset pre-empts mistakes

```
index.is - vs-code-can-do-that
        index.js
                rick = () \Rightarrow \{
                   var messages = [
                     'Never gonna give you up',
                     'Never gonna let you down',
                     'Never gonna run around and desert you'
                   ];
                   for (i = 0; i \neq messages.length; i++) {
8
                     console.log(messages[i]);
          10
¢
                }; <sup>±</sup>
          11
          12
                rick();
          13
          14
办
Ø 0 A 0
                  Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript ESLint!
                                                                       Prettier: <
```

Linting > Debugging

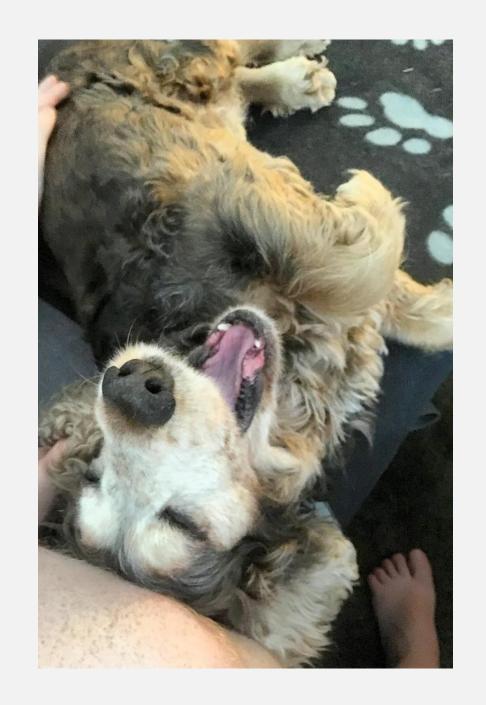
- Find mistakes while you make them
- Based on the experience and consensus of many others
- Learn from explanations of linting results
- Install and configure, or use in-built linting.



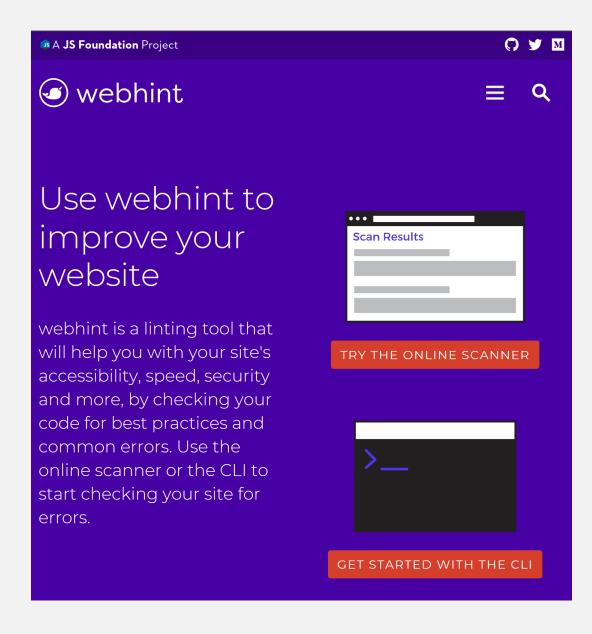
code.visualstudio.com

Moving from knowing .*

to
learning by making mistakes



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



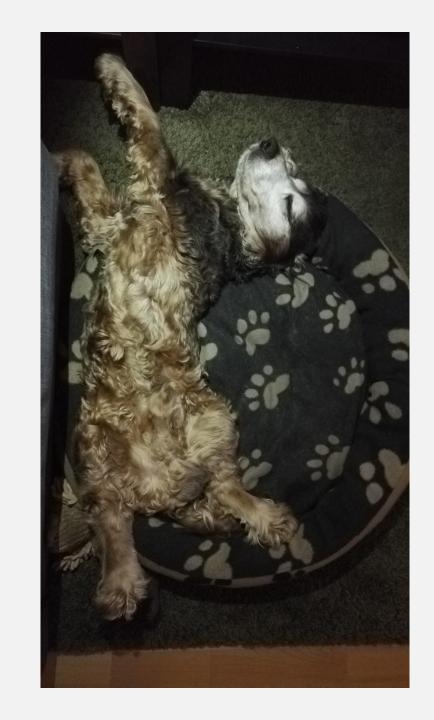
webhint.io

Getting to know our tooling

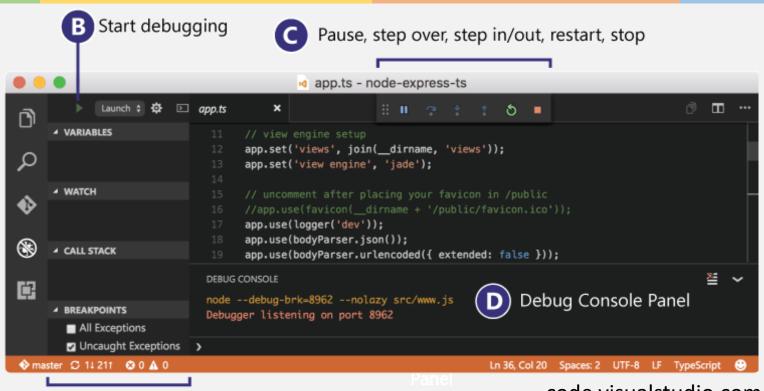
It is true that our tools have become much more complex.



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



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



Debug Side Bar

code.visualstudio.com/Docs/editor/debugging

Giving back to others

Take part in the JavaScript community...

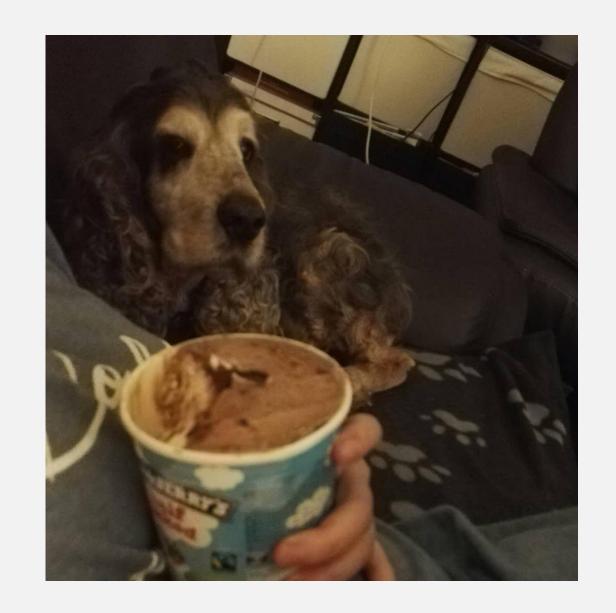
- Help document
- Help clean up and send pull requests
- Publish your own work as open source
- Enjoy meetups and events
- Help by example, not by telling people what they should do

Muffling the noise...

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



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++) {</pre>
        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--) {
        if (array[all] > temp) {
            temp = array[all];
    return temp;
```

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

Using native methods instead of "if"...

```
const largestInArray = function(array) {
   let temp = 0;
   let all = array.length;
   while (all--) {
      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 <u>Terminal 5 Insider</u> Editor photo by <u>John Oxley Library</u> Browser photo by <u>Joseph Brent</u>

