CSS Isn't Scary

@sublimemarch

Stephanie Slattery
@sublimemarch

Raise your hand if…

@sublimemarch

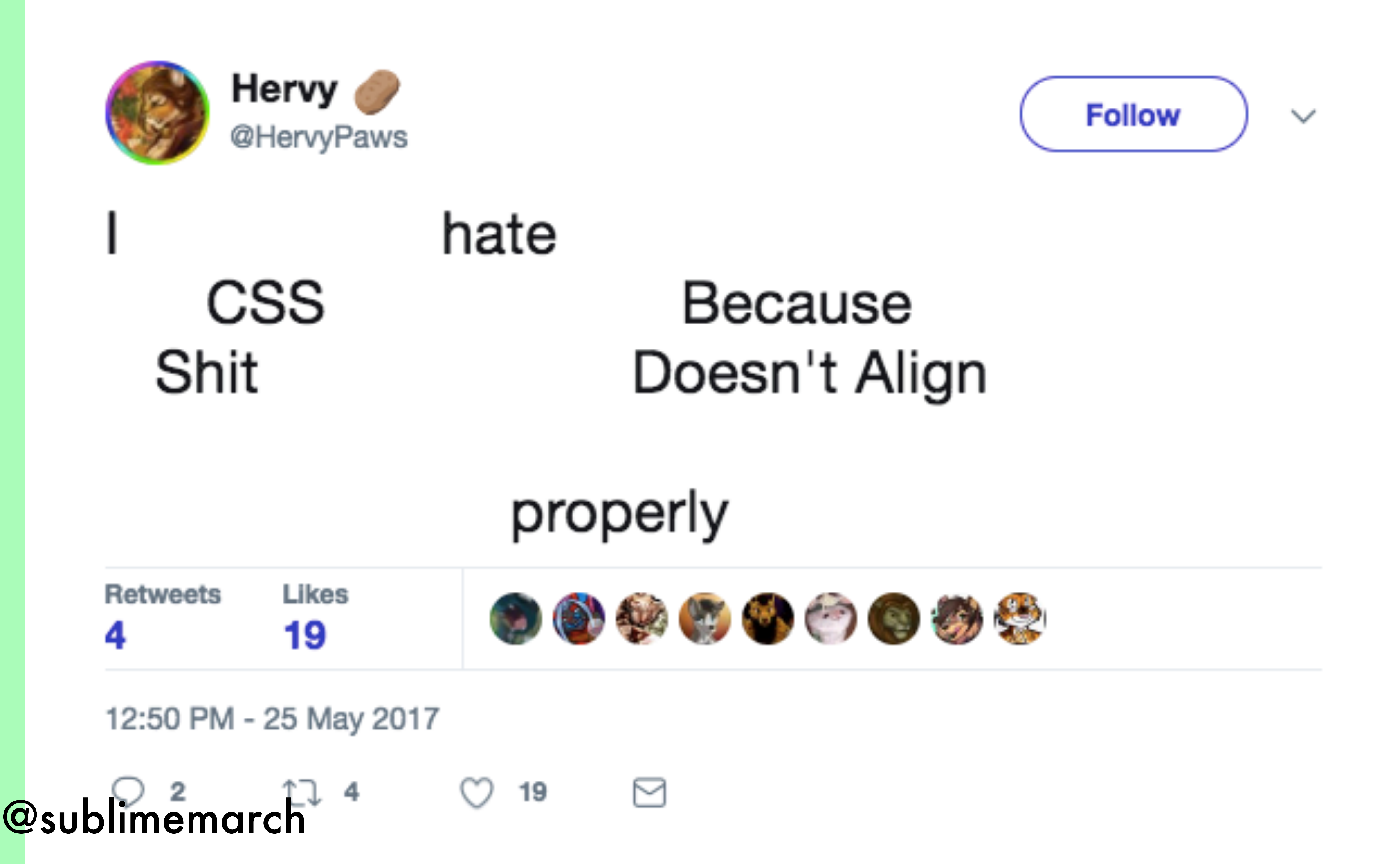
"CSS is strangely considered both one of the easiest and one of the hardest languages to learn as a web developer."

1. Why CSS is scary
2. Why CSS is great!
3. How to write better CSS

1. Why CSS is scary
2. Why CSS is great!
3. How to write better CSS

@sublimemarch

@sublimemarch



Hervy
@HervyPaws

Follow

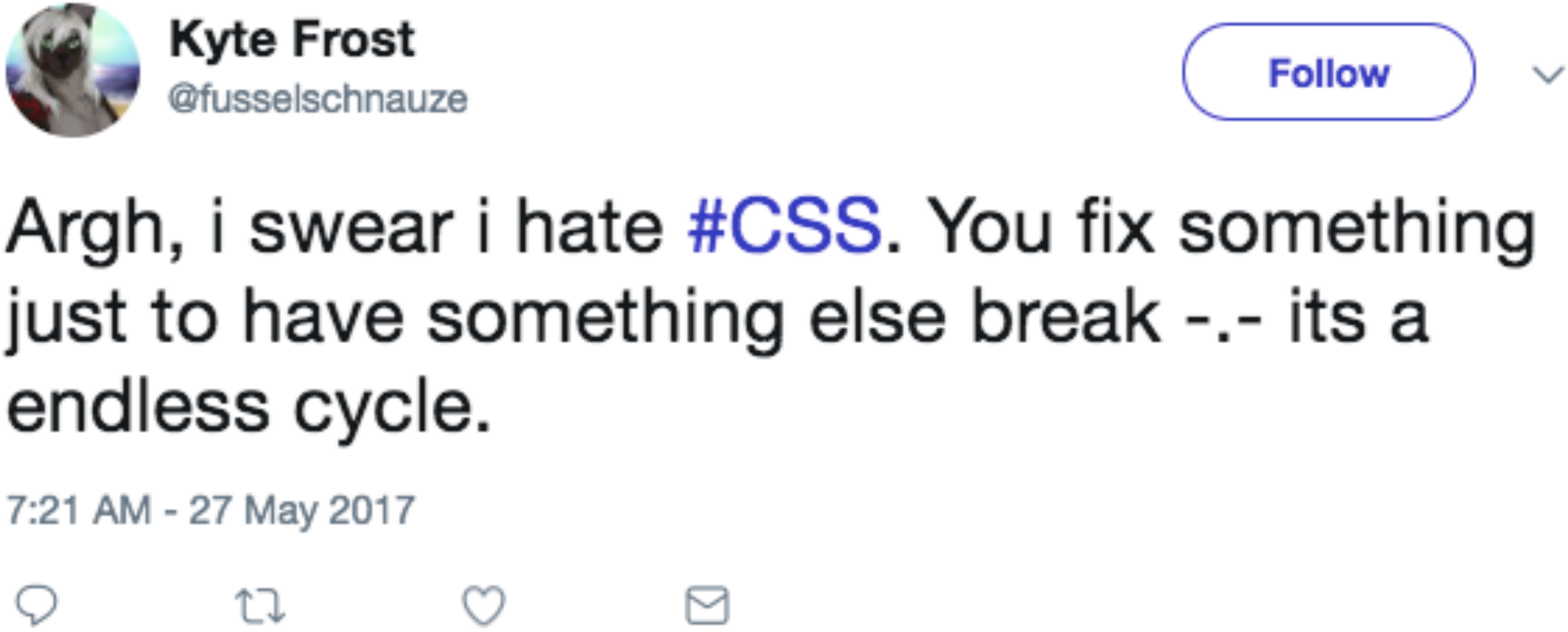
I hate
CSS Because
Shit Doesn't Align

properly

Retweets 4 Likes 19

12:50 PM - 25 May 2017

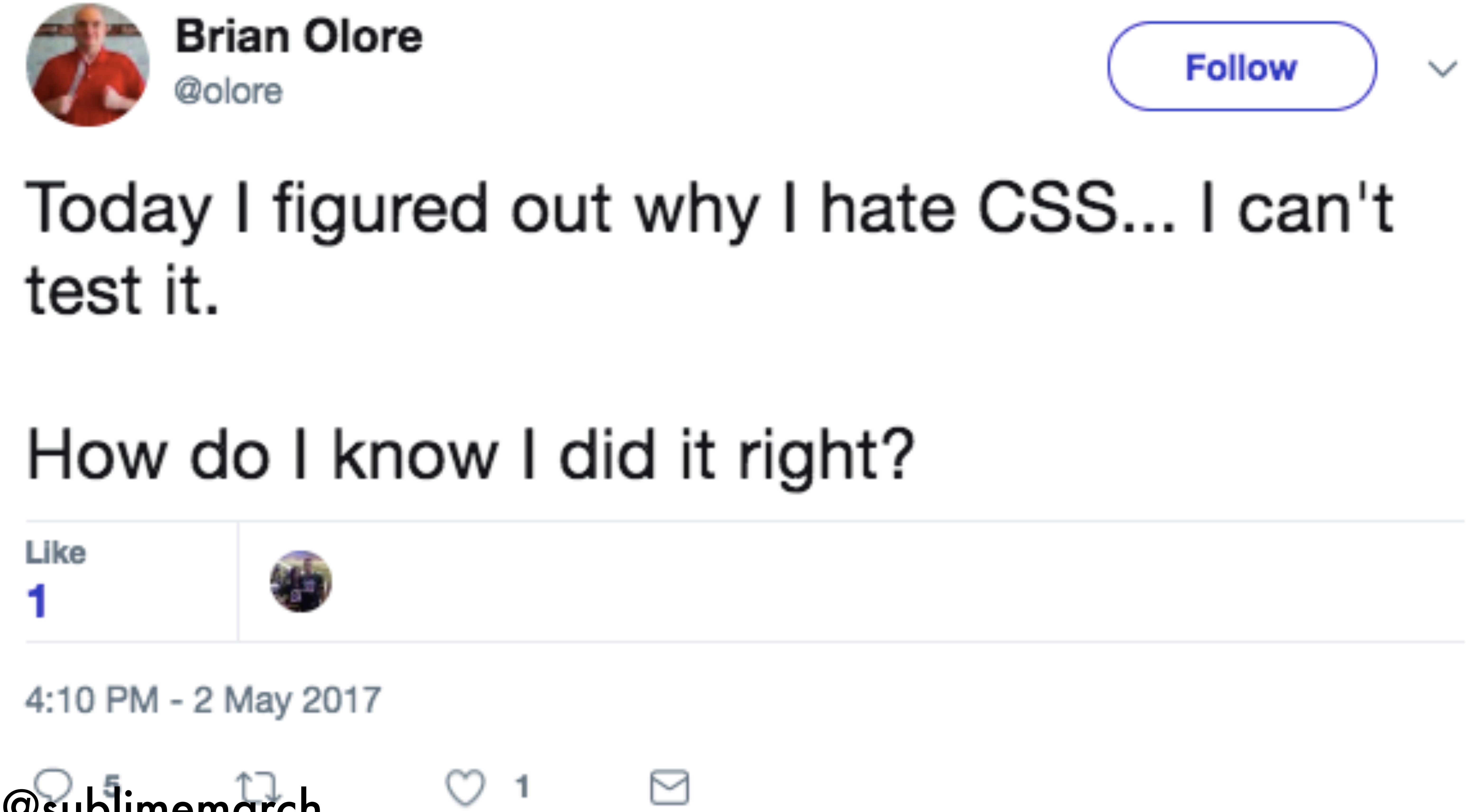
2 4 19



Kyte Frost
@fusselschnauze

Argh, i swear i hate #CSS. You fix something just to have something else break -.- its a endless cycle.

7:21 AM - 27 May 2017



Brian Olore
@olore

Today I figured out why I hate CSS... I can't test it.

How do I know I did it right?

Like
1

4:10 PM - 2 May 2017

@sublimemarch

CSS is declarative.

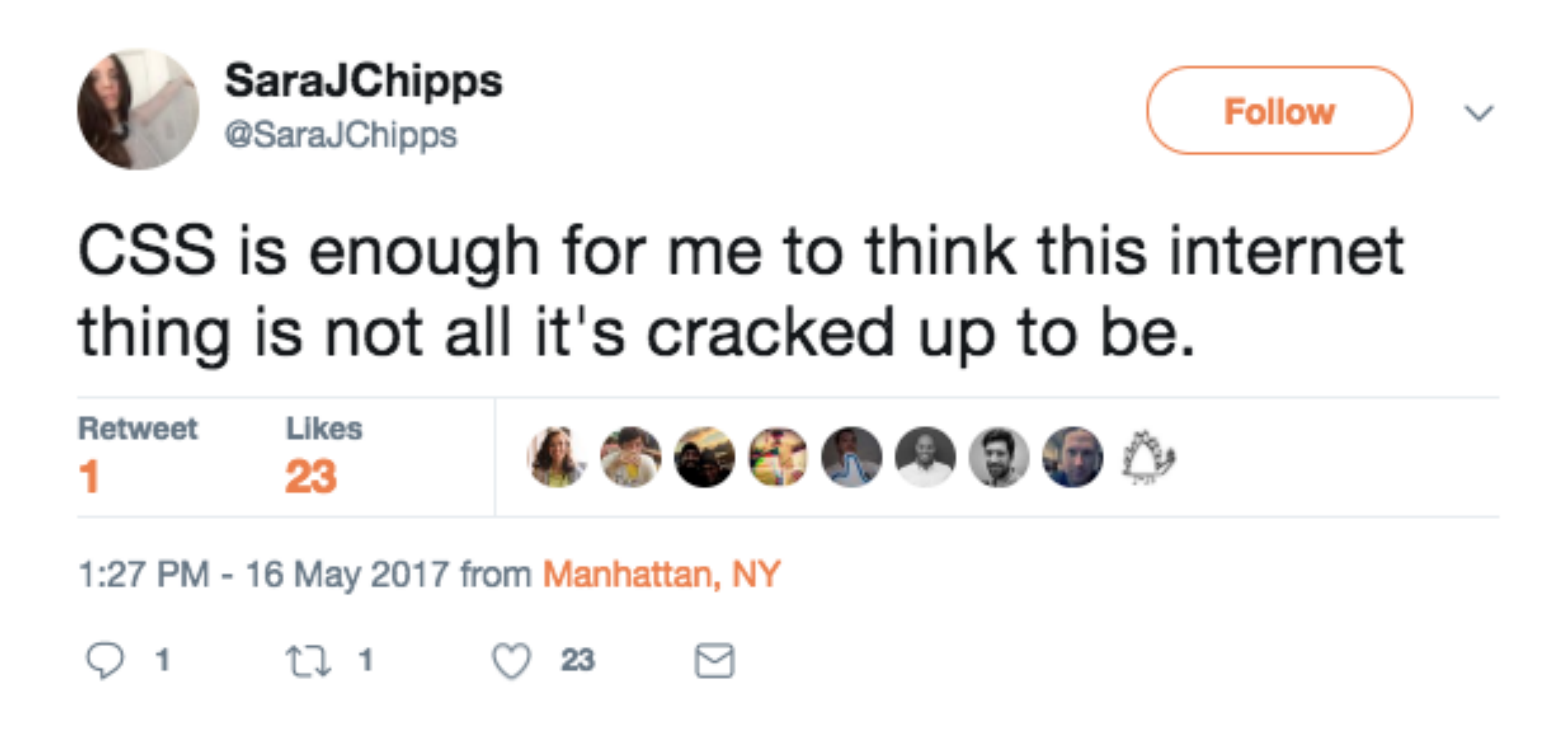
In this declarative language...

✦ the last rule declared takes precedence

✦ the rule declared on the most specific selector takes precedence

✦ there's no such thing as scope - everything is global!

@sublimemarch



SaraJChipps
@SaraJChipps

CSS is enough for me to think this internet thing is not all it's cracked up to be.

Retweet 1 Likes 23

1:27 PM - 16 May 2017 from Manhattan, NY

CSS IS *awful!*

@sublimemarch

CSS IS *awesome!*

1. Why CSS is scary
2. Why CSS is great!
3. How to write better CSS

@sublimemarch

The Separation of Controls Principle

"A design principle for separating a computer program into distinct sections, such that each section addresses a separate concern. A concern is a set of information that affects the code of a computer program." - Wikipedia

We separate concerns into HTML, CSS, and Javascript.

@sublimemarch

HTML organizes content.

CSS defines presentation.

JS defines how content interacts and behaves with the user.

@sublimemarch

CSS is flexible.

CSS is made of simple things.

@sublimemarch

```
.selector {
  name: value;
}
```

CSS is easy to generate.

@sublimemarch

CSS stands alone.

Just look at this! Or this!

CSS is open source.

✨ CSS tries its best. ✨

@sublimemarch

1. Why CSS is scary
2. Why CSS is great!
3. How to write better CSS

CSS stops being scary when you understand it and follow best practices.

CSS is just another programming language

@sublimemarch

Apply your programming skills:

- Reading the docs
- Planning your code
- Pseudocoding
- Refactoring

Understand specificity

Understand specificity

0. Inline styles

Understand specificity

0. Inline styles
1. IDs

Understand specificity

```
0. Inline styles
1. IDs
2. Classes, attributes, and pseudo-classes
            [type="radio"]    :hover
```

@sublimemarch

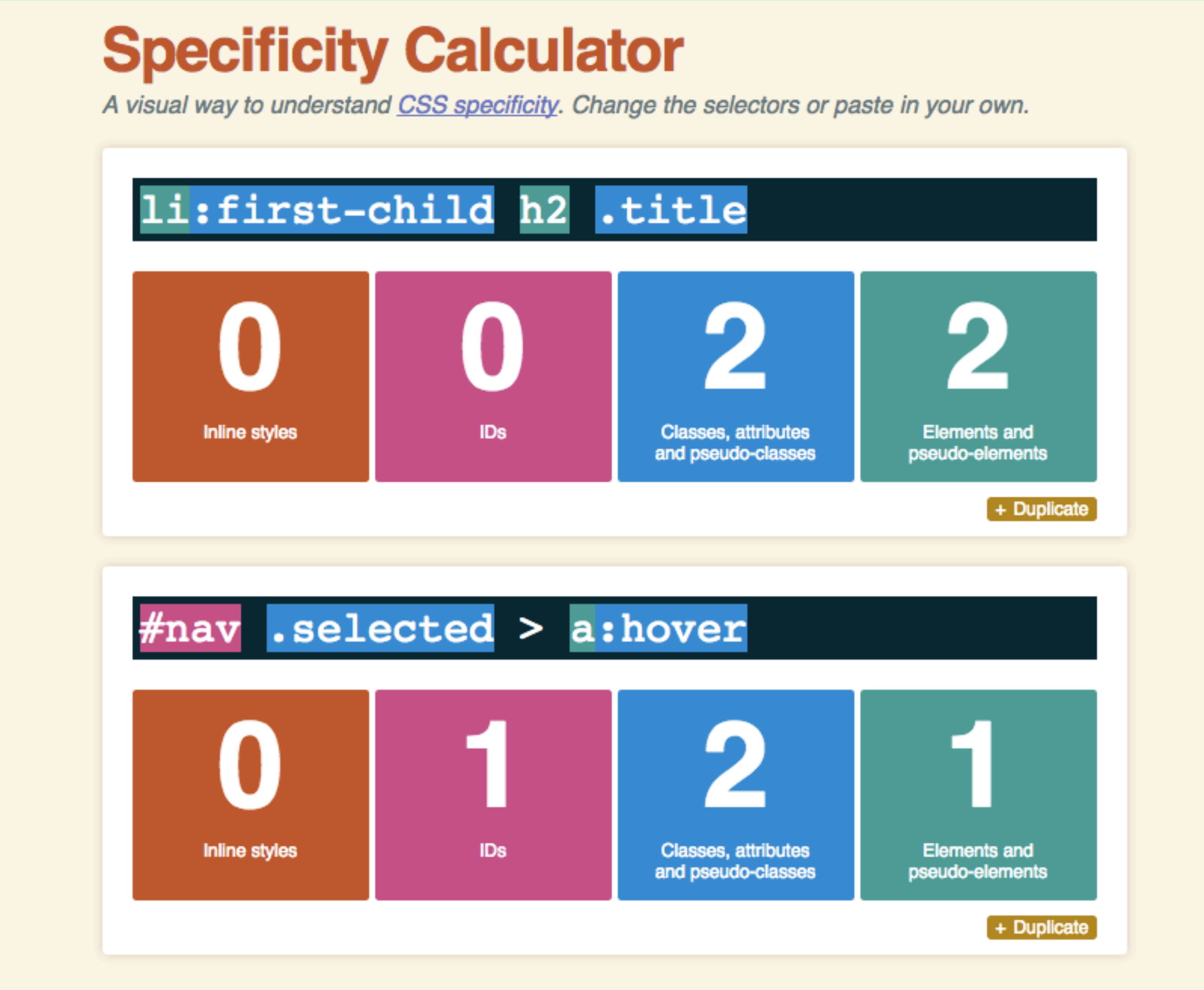
Understand specificity

0. Inline styles
1. IDs
2. Classes, attributes, and pseudo-classes
3. Elements and pseudo-elements

Understand specificity

```
0. Inline styles
1. IDs
2. Classes, attributes, and pseudo-classes
3. Elements and pseudo-elements
      h1          :before
```

Don't guess and check for specificity!



Specificity Calculator

A visual way to understand CSS specificity. Change the selectors or paste in your own.

```
li:first-child h2 .title
```

Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements
0	0	2	2

```
#nav .selected > a:hover
```

Inline styles	IDs	Classes, attributes and pseudo-classes	Elements and pseudo-elements
0	1	2	1

http://specificity.keegan.st/

@sublimemarch

Don't over-specify

```css
#home #hero #claim .logo h2 {
  display: inline-block;
}
```

Don't over-specify

```
#home #hero #claim .logo h2 {
  display: inline-block;
}
```

For any h2 inside anything with the logo class that's inside of the claim element that's inside of the hero element that's inside of the home element, display with inline-block.

Very specific selectors are hard to override in the future.

No !important flags

```
<p id="pink-text">Kittens are cute.</p>
```

@sublimemarch

No !important flags

```
<p id="pink-text">Kittens are cute.</p>

#pink-text {
  color: pink;
}
```

No !important flags

```
<p id="pink-text">Kittens are cute.</p>

#pink-text {
  color: pink;
}

p {
  color: black !important;
}
```

And no inline styles

```html
<div style="color: pink;">I love kittens.</div>
```

Use a single class as your selector

```css
.hero-text-link {
  font-size: 18px;
}
```

Use a single class as your selector

```css
.hero-text-link {
  font-size: 18px;
}
```

instead of something more complex

```css
.hero p a {
  font-size: 18px;
}
```

@sublimemarch

Keep it DRY

Keep it DRY

(don't repeat yourself)

```
<h2 class="fun-title pink-title">Hello</h2>

.fun-title {
  font-family: "Comic Sans", sans-serif;
}

.pink-title {
  font-family: "Comic Sans", sans-serif;
  color: pink;
}
```

instead

```
<h2 class="fun-title pink-title">Hello</h2>

.fun-title, .pink-title {
  font-family: "Comic Sans", sans-serif;
}

.pink-title {
  color: pink;
}
```

or even better

```
<h2 class="title pink-title">Hello</h2>

.title {
  font-family: "Comic Sans", sans-serif;
}

.pink-title {
  color: pink;
}
```

or even better-er

```
<h2 class="title pink">Hello</h2>

.title {
  font-family: "Comic Sans", sans-serif;
}

.pink.title {
  color: pink;
}
```

@sublimemarch

CSS extensions make this even easier!

@sublimemarch

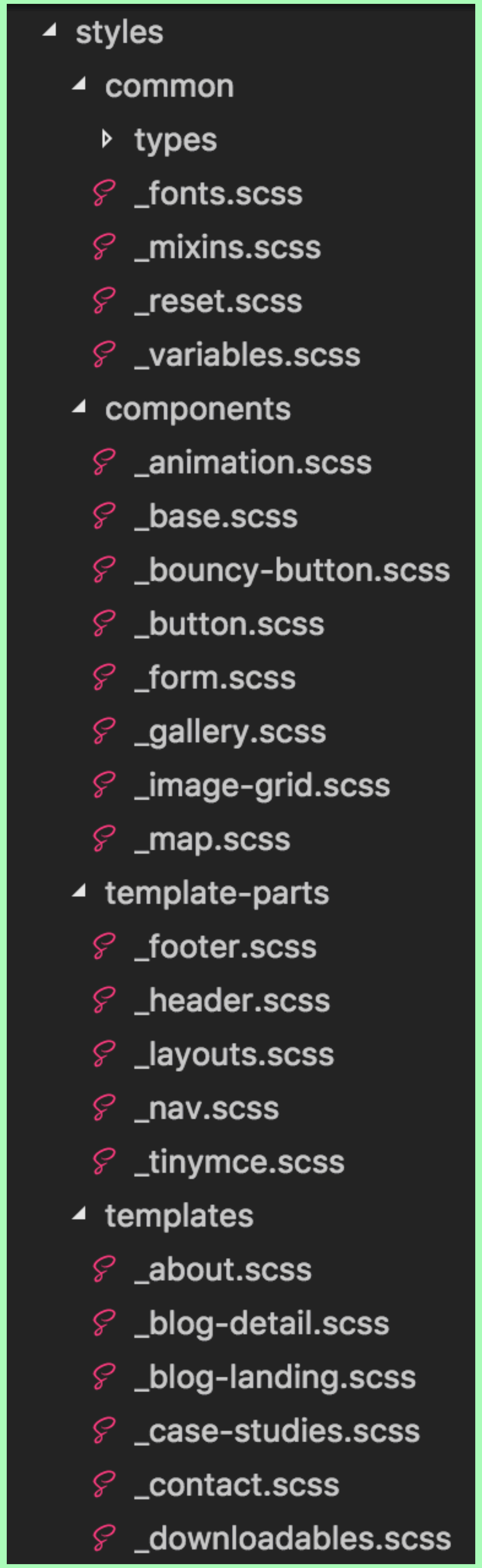
```
<h2 class="pink-title">Hello!</h2>

.title {
  font-family: "Comic Sans", sans-serif;
}

.pink-title {
  @extend .title;
  color: pink;
}
```

@sublimemarch

Organize your CSS.



```
styles
  common
    types
    _fonts.scss
    _mixins.scss
    _reset.scss
    _variables.scss
  components
    _animation.scss
    _base.scss
    _bouncy-button.scss
    _button.scss
    _form.scss
    _gallery.scss
    _image-grid.scss
    _map.scss
  template-parts
    _footer.scss
    _header.scss
    _layouts.scss
    _nav.scss
    _tinymce.scss
  templates
    _about.scss
    _blog-detail.scss
    _blog-landing.scss
    _case-studies.scss
    _contact.scss
    _downloadables.scss
```

Organize your files, but also your selectors.

hacks.css

In hacks.css, you should leave:

- your hacky code
- why you did it
- possible ways to fix it

Put it in a hacks.css file if you're

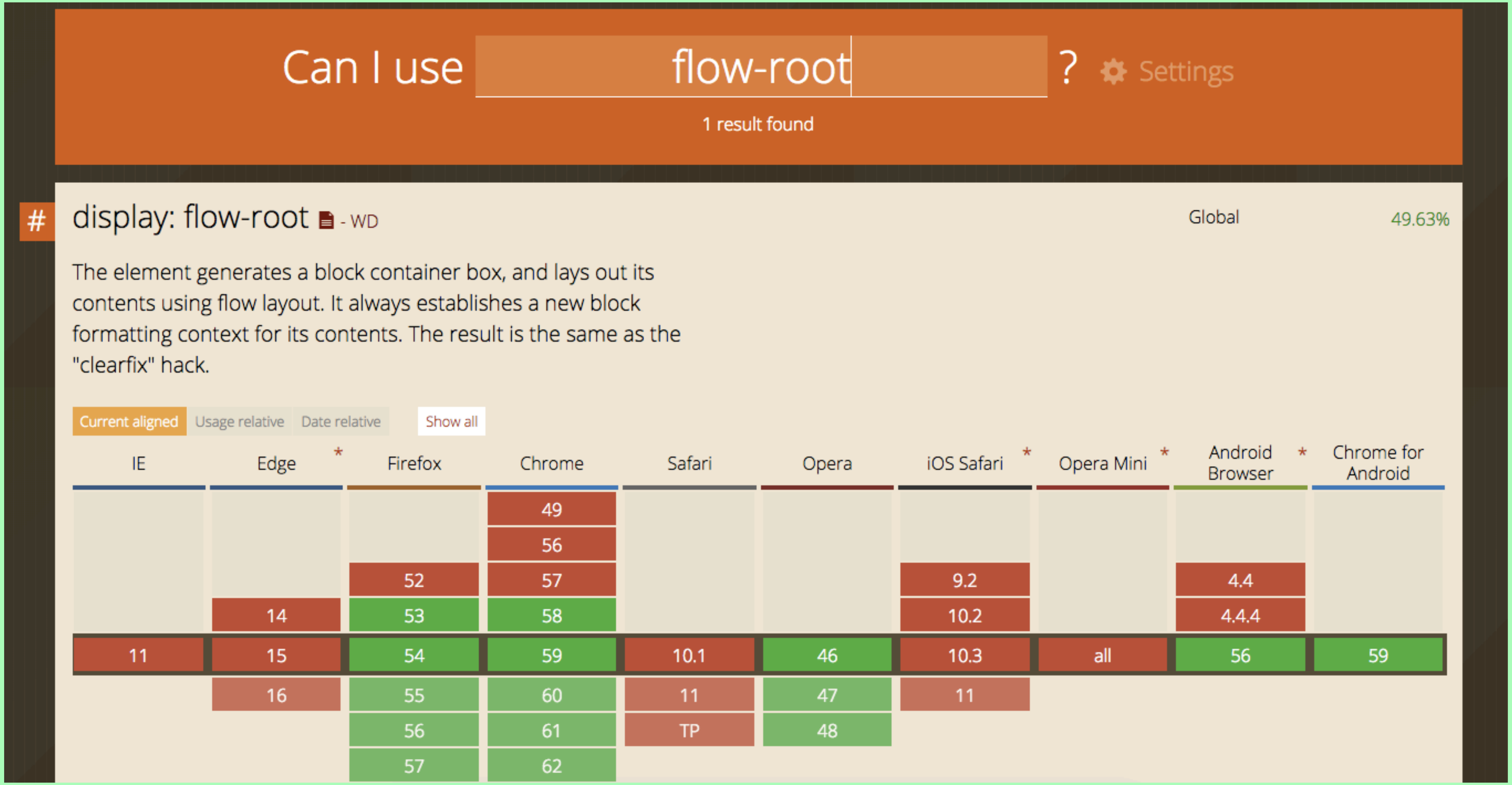
✦ using magic numbers

✦ writing overly specific selectors

✦ using !important flags

✦ undoing styles that are elsewhere in the code

Understand browser compatibility.



Can I use flow-root ? Settings

1 result found

display: flow-root - WD

Global 49.63%

The element generates a block container box, and lays out its contents using flow layout. It always establishes a new block formatting context for its contents. The result is the same as the "clearfix" hack.

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56						
		52	57			9.2		4.4	
	14	53	58			10.2		4.4.4	
11	15	54	59	10.1	46	10.3	all	56	59
	16	55	60	11	47	11			
		56	61	TP	48				
		57	62						

https://caniuse.com/

@sublimemarch

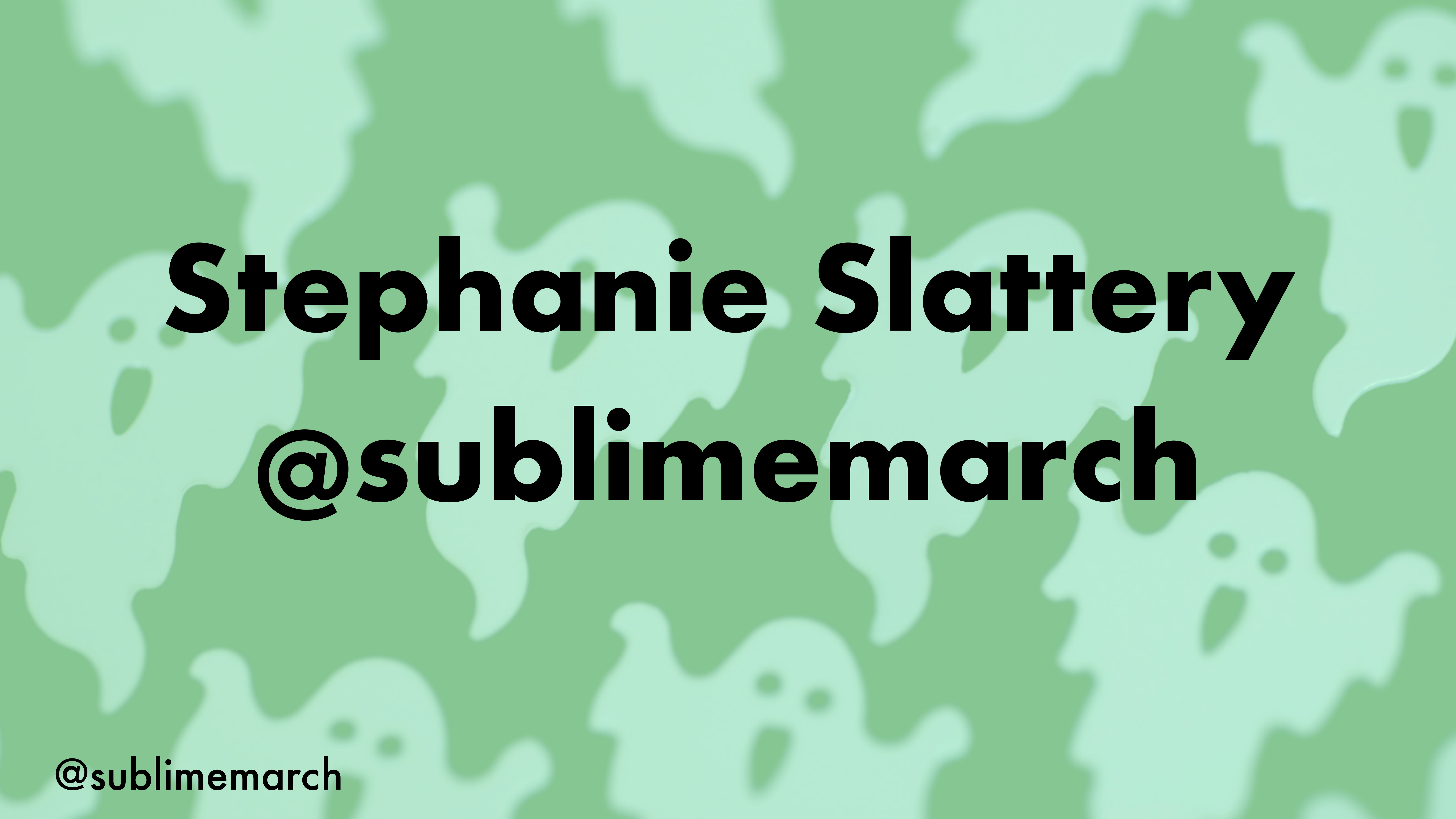
And so much more!

- learn the box model
- use flexbox
- pick a preprocessor
- implement a naming methodology like BEM or OOCSS
- use a linter
- look at dev tools
- use a CSS reset

CSS IS *awesome*!

(mostly)

@sublimemarch



Stephanie Slattery
@sublimemarch