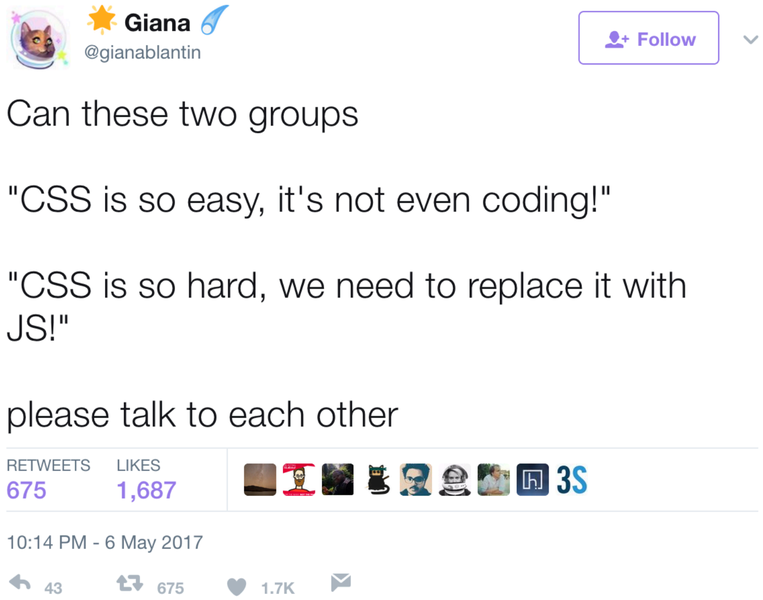
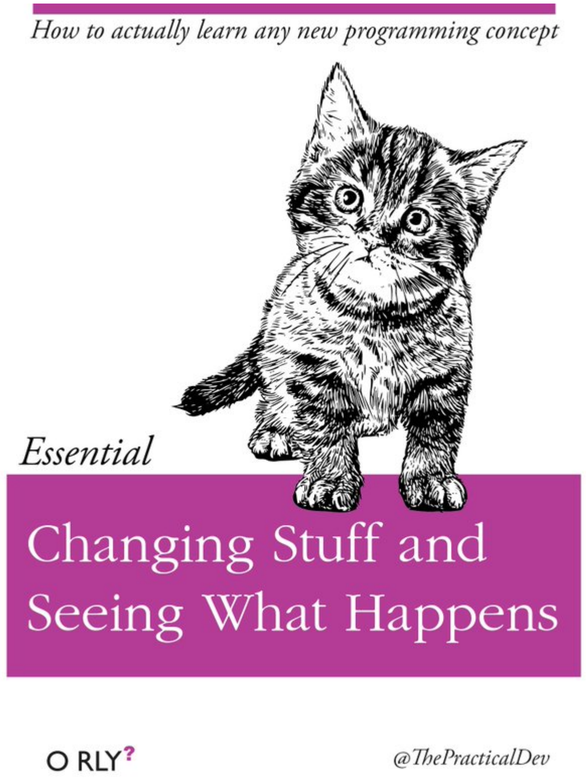
CSS vs. JavaScript: Trust vs. Control

Chris Heilmann
@codepo8
christianheilmann.com


PICK A SIDE
MARVEL
CIVIL WAR
CAPTAIN AMERICA

CSS seems to be confusing for programmers…







https://twitter.com/gianablantin/status/860965927898054656

https://twitter.com/floatvoid/status/863605150799118336

CSS is fault tolerant – this can be confusing

JavaScript is not fault tolerant – this can be disastrous

CSS

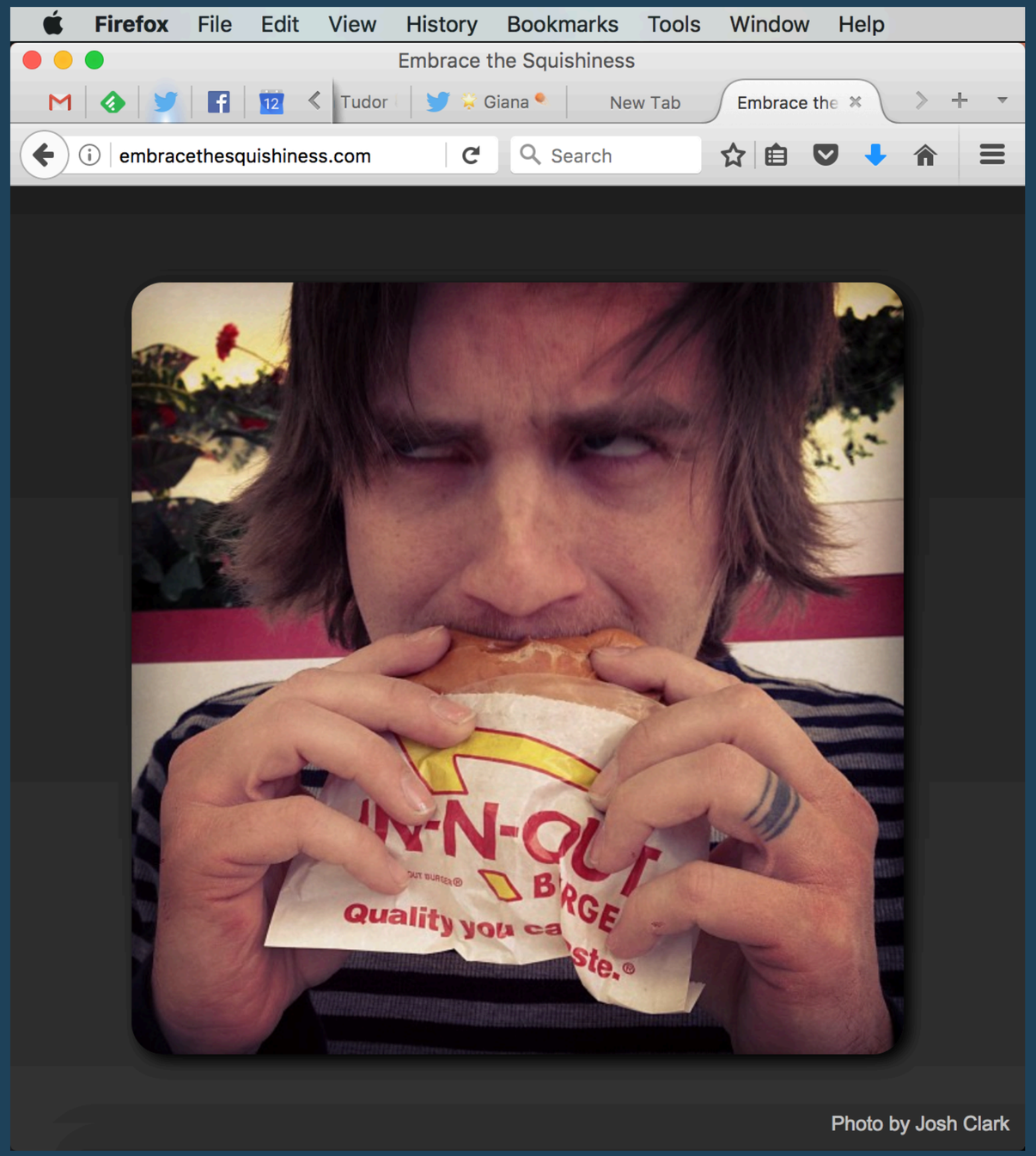
You apply your styles and you hope it worked.

JavaScript

You control the styling and you can and should verify that it worked

CSS

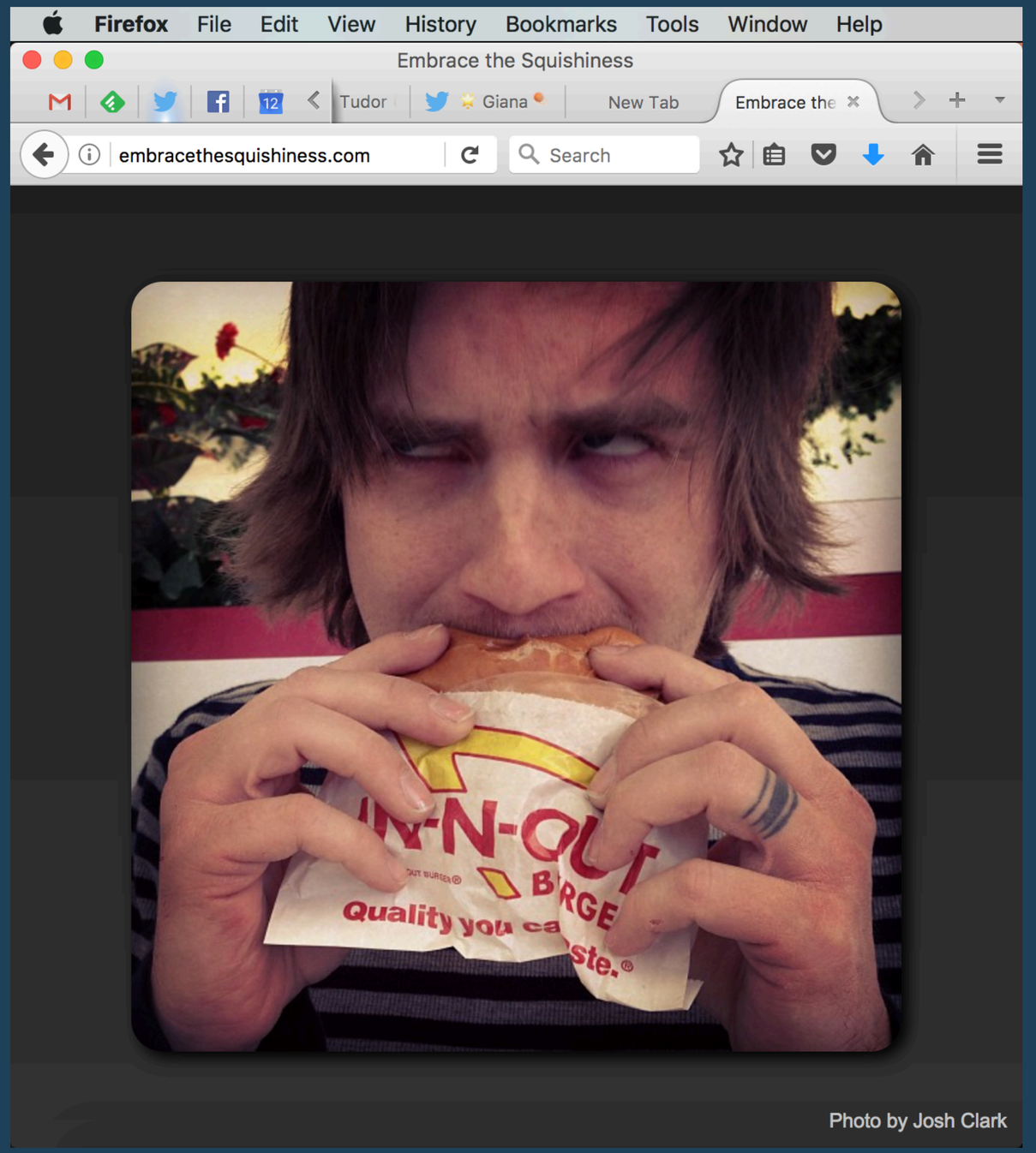
Means embracing the squishiness of the web!



embracethesquishiness.com

Things beyond your control:

- The browsers of your users
- The resolution, pixel depth and colour settings of their devices
- Their connection reliability and speed
- Their connection restrictiveness – resources may be blocked
- Their font size and zoom needs
- The availability of resources on their machines for your product (is the CPU already burning?)
- The amount of text content and image sizes in your product – CMS anyone?



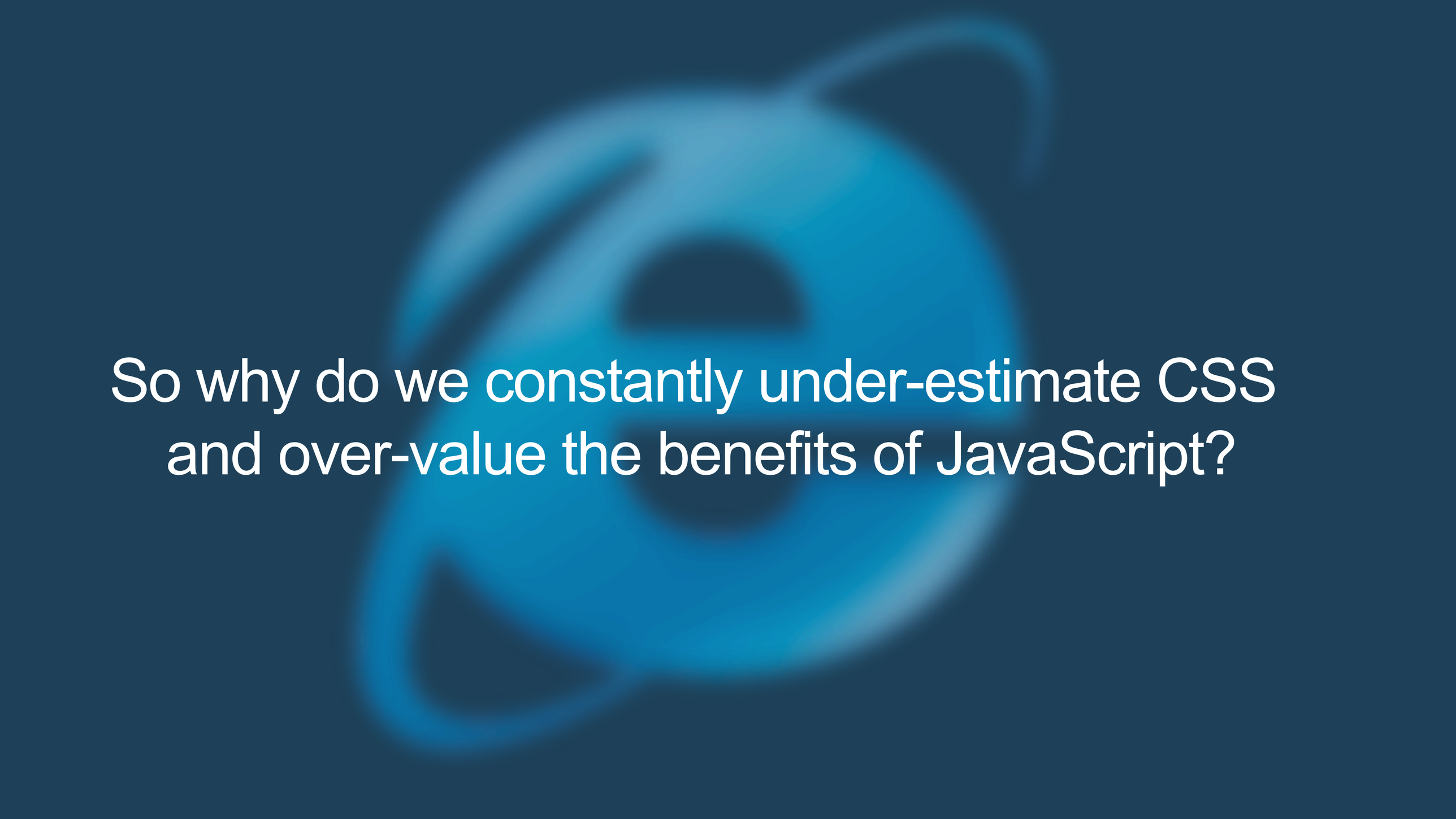
embracethesquishiness.com

CSS

It is the job of the browser to perform well, use GPU resources and skip functionality.

JavaScript

It’s your job to test for support, to ensure rendering, painting and reflow is fast and to keep animation in sync.


So why do we constantly under-estimate CSS
and over-value the benefits of JavaScript?

CSS had a bumpy history…

- Initially very limited and a replacement for visual HTML and attributes (font, bgcolor, align…)
- Patchy browser support and very odd errors without debugging options
- Massive jump in functionality when iPhone became a thing and Apple called the shots
- Lots of problems with browser wars around prefixes
- Up until recently no layout model, but hacks using positioning and floating. So, too restrictive or too confusing

Our solution was to patch with JavaScript.

We can read out conditions and react to them creating HTML and applying styling.

But CSS has been going leaps and bounds

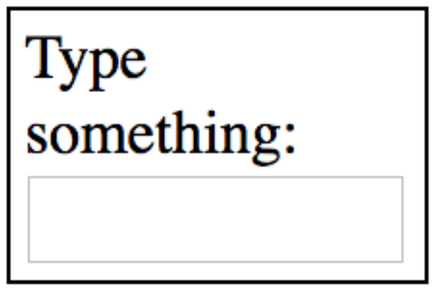
- Evergreen browsers are a thing
- Browser tooling gives detailed insights
- Support is well documented: caniuse.com
- Support channels and bug tracking is available for almost all browsers
- Pre-processors like Sass and Less have turned up the heat to innovate the spec faster, much like jQuery inspired JavaScript of today
- Approaches like OOCSS (Nicole Sullivan) and Atomic Design (Brad Frost) help work with maintainability across different componentss

So here are some amazing things CSS can do now and you should consider using…

Calculated CSS values

Calculations

```
1 <form>
2   <div id="formbox">
3   <label>Type something:</label>
4   <input type="text">
5   </div>
6 </form>
```



developer.mozilla.org/en/docs/Web/CSS/calc

Calculations

```
input {
  padding: 2px;
  display: block;
  width: 98%;                    /* fallback for browsers without support
  width: calc(100% - 1em);
}

#formbox {
  width: 130px;                  /* fallback for browsers without support
  width: calc(100% / 6);
  border: 1px solid black;
  padding: 4px;
}
```

developer.mozilla.org/en/docs/Web/CSS/calc

Calculations

calc() as CSS unit value - CR

Global 92.31% + 1.78% = 94.09%

unprefixed: 92.1% + 1.78% = 93.88%

Method of allowing calculated values for length units, i.e. `width: calc(100% - 3em)`

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						
			57			9.2		1 4.4	
	14	52	58			10.2		1 4.4.4	
11	15	53	59	10.1	45	10.3	all	56	59
	16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

developer.mozilla.org/en/docs/Web/CSS/calc

Media Queries

Media Queries

```
<!-- CSS media query on a link element -->
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />

<!-- CSS media query within a stylesheet -->
<style>
@media (max-width: 600px) {
  .facet_sidebar {
    display: none;
  }
}
</style>
```

developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Media Queries

CSS3 Media Queries - REC

Global 97.77% + 0.01% = 97.77%

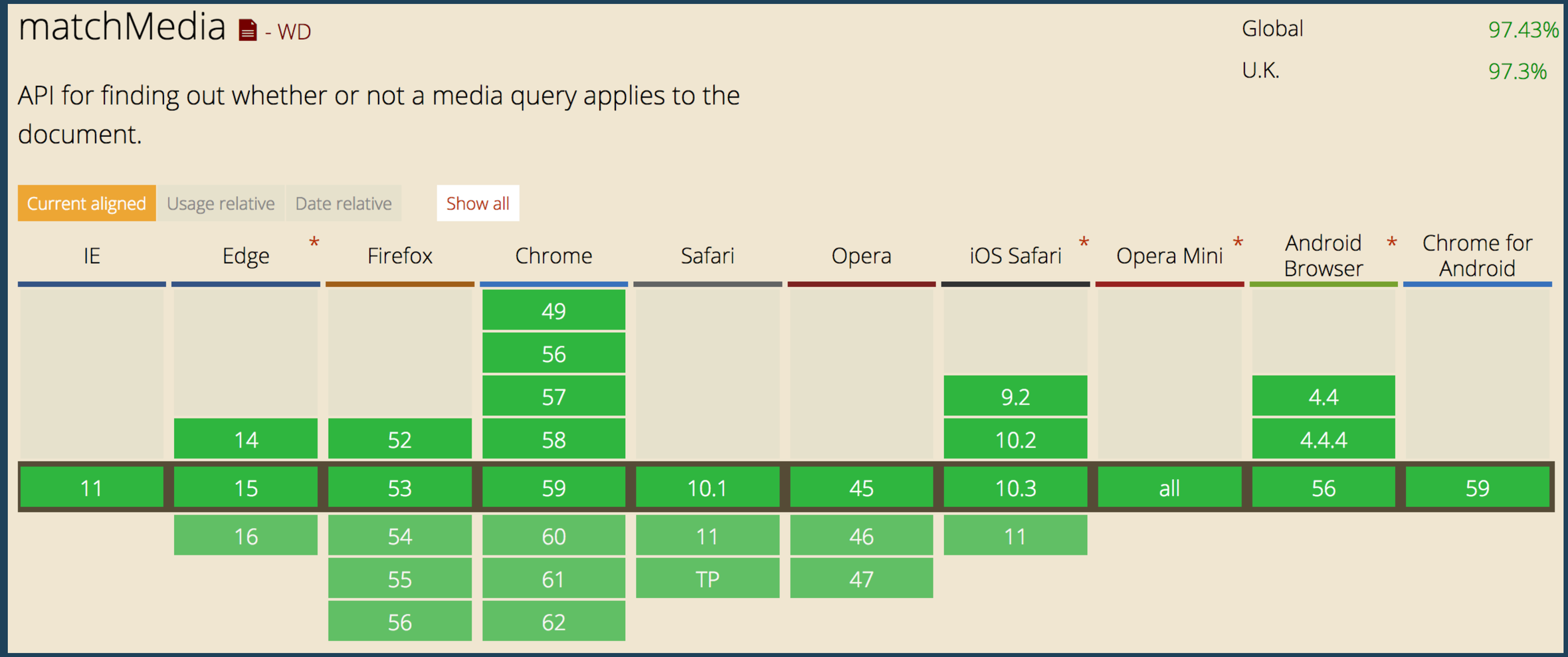
Method of applying styles based on media information. Includes things like page and device dimensions

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						
			57			9.2		4.4	
	14	52	58			10.2		4.4.4	
1 11	15	53	59	10.1	45	10.3	all	56	59
	16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Media Queries



developer.mozilla.org/en-US/docs/Web/API/Window/matchMedia

Generated content

Generated Content

HTML content

```
1 | <q>Some quotes</q>, he said, <q>are better than none</q>.
```

CSS content

```
1 | q::before {
2 |   content: "«";
3 |   color: blue;
4 | }
5 | q::after {
6 |   content: "»";
7 |   color: red;
8 | }
```

Output

«Some quotes», he said, «are better than none».



https://developer.mozilla.org/en-US/docs/Web/CSS/::before

a.singlediv.com

Generated Content

CSS Generated content for pseudo-elements - REC

Global 97.81% + 0.35% = 98.15%

Method of displaying text or images before or after the given element's contents using the ::before and ::after pseudo-elements. All browsers with support also support the `attr()` notation in the `content` property.

Current aligned | Usage relative | Date relative | Show all

http://a.singlediv.com/

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56						
			57			9.2		4.4	
	14	52	58			10.2		4.4.4	
11	15	53	59	10.1	45	10.3	all	56	59
	16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

Animations and transitions

Animations / Transitions

```css
p {
  animation-duration: 3s;
  animation-name: slidein;
}

@keyframes slidein {
  from {
    margin-left: 100%;
    width: 300%;
  }

  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

```css
#foo {
  border-radius: 50px;
  width: 50px;
  height: 50px;
  background: #c00;
  position: absolute;
  top: 0;
  left: 0;
  transition: transform 1s;
}
```

```js
var f = document.getElementById('foo');
document.addEventListener('click', function(ev){
    f.style.transform = 'translateY('+(ev.clientY-25)+'px)';
    f.style.transform += 'translateX('+(ev.clientX-25)+'px)';
},false);
```

developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations

developer.mozilla.org/en-US/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions

Animations / Transitions

CSS Animation - WD

Complex method of animating certain properties of an element

Global	94.67% + 0.03% =	94.71%
unprefixed:	82.23%	
U.K.	97.02% + 0.04% =	97.06%
unprefixed:	93.99%	

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						
			57			9.2		4.4	
	14	52	58			10.2		4.4.4	
11	15	53	59	10.1	45	10.3	all	56	59
	16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

Animations / Transitions

```
function listener(e) {
  var l = document.createElement("li");
  switch(e.type) {
    case "animationstart":
      l.innerHTML = "Started: elapsed time is " + e.elapsedTime;
      break;
    case "animationend":
      l.innerHTML = "Ended: elapsed time is " + e.elapsedTime;
      break;
    case "animationiteration":
      l.innerHTML = "New loop started at time " + e.elapsedTime;
      break;
  }
  document.getElementById("output").appendChild(l);
}
```

Viewport Units

Viewport Units

Viewport Width (vw)

A percentage of the full viewport width.
f.e. 480px wide screen - 10vw is 10% or 48px

% is percentage of the parent element, vw is percentage of the window.

Viewport Height (vh)

A percentage of the full viewport height.

Viewport Minimum (vmin)

A percentage of the viewport width or height, whichever is smaller.

10vmin = 10% of the current viewport width in portrait and 10% of the viewport height on landscape.

Viewport Maximum (vmax)

A percentage of the viewport width or height, whichever is larger.

Viewport Units

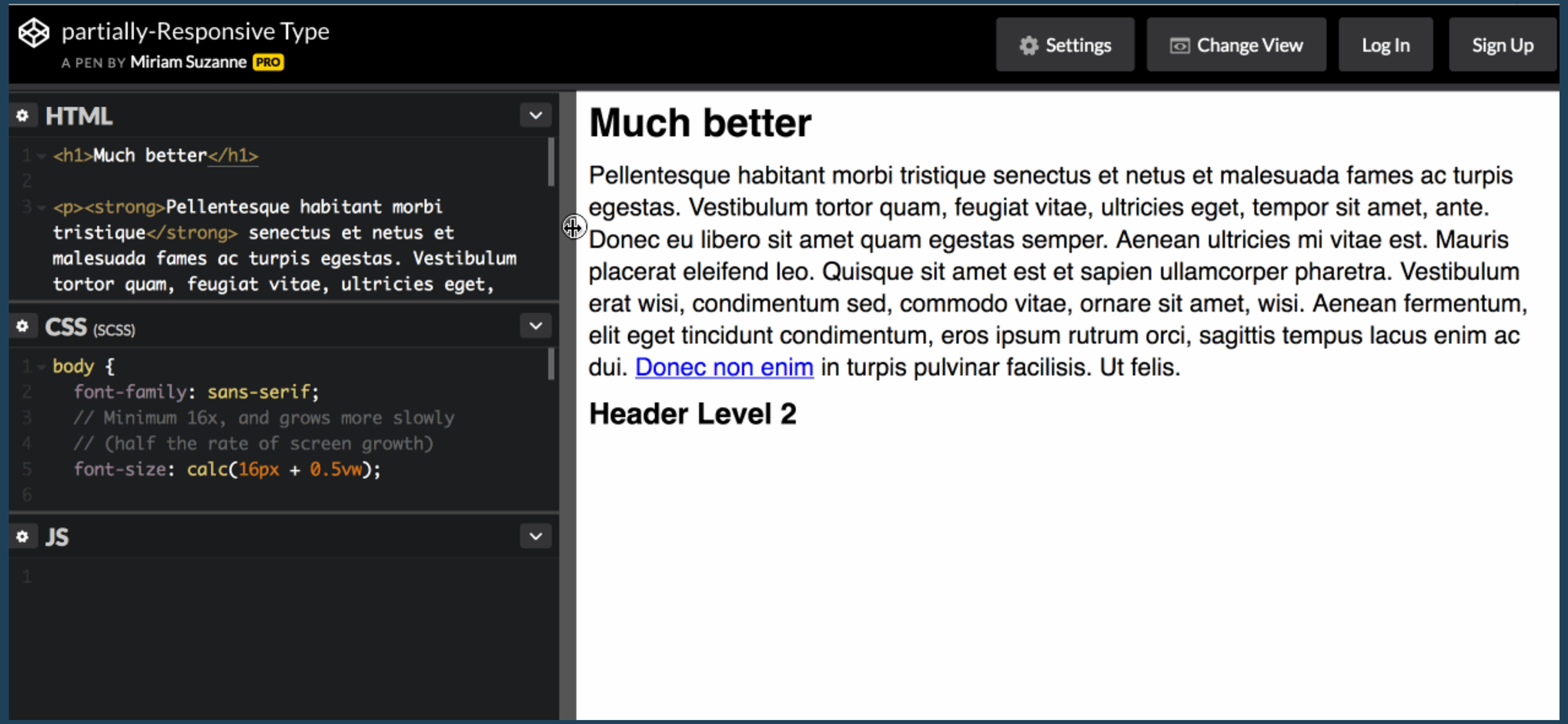
```
body {
  // font grows 1px for every 100px of viewport width
  font-size: calc(16px + 1vw);
  // leading grows along with font,
  // with an additional 0.1em + 0.5px per 100px of the viewport
  line-height: calc(1.1em + 0.5vw);
}
```

```
h1 {
  font-size: calc(1.2em + 3vw);
}

@media (min-width: 50em) {
  h1 {
    font-size: 50px;
  }
}
```

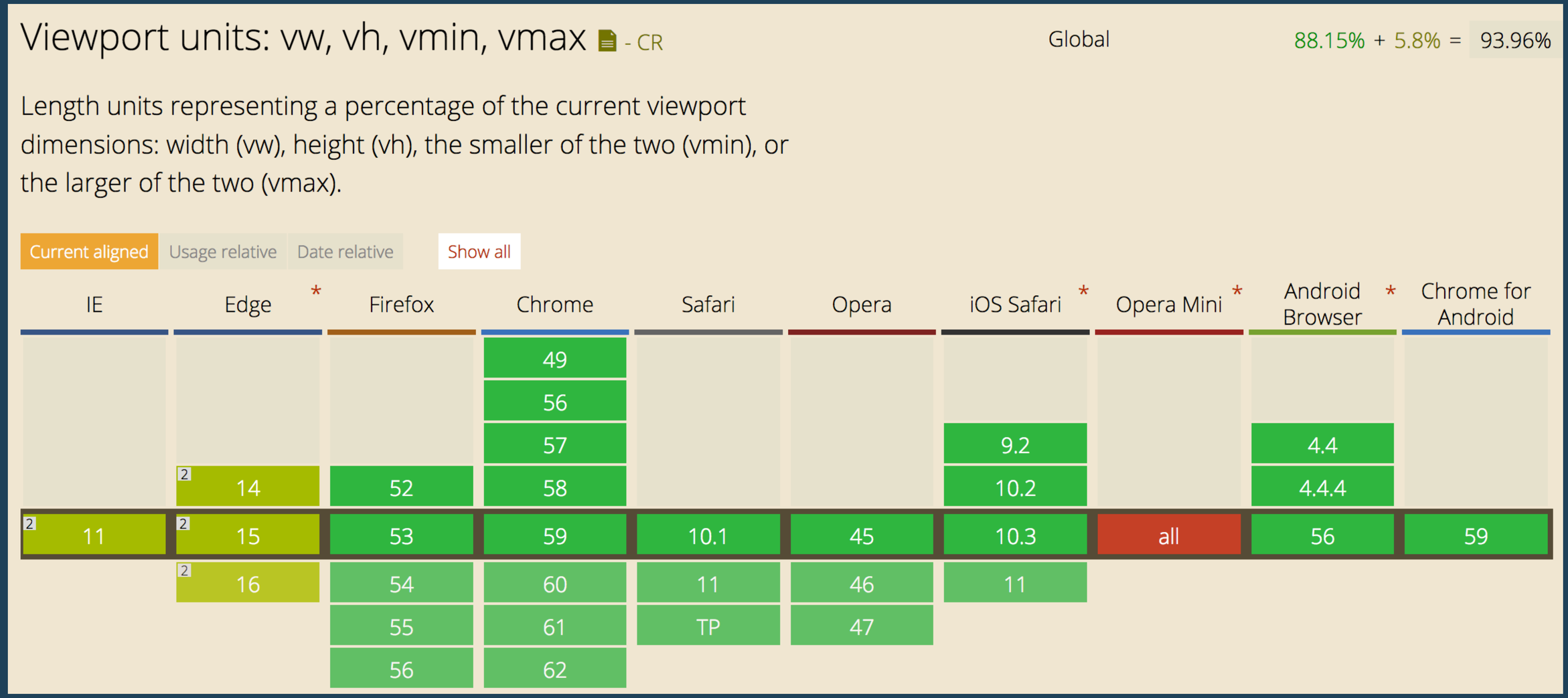
css-tricks.com/fun-viewport-units/

Viewport Units



css-tricks.com/fun-viewport-units/

Viewport Units



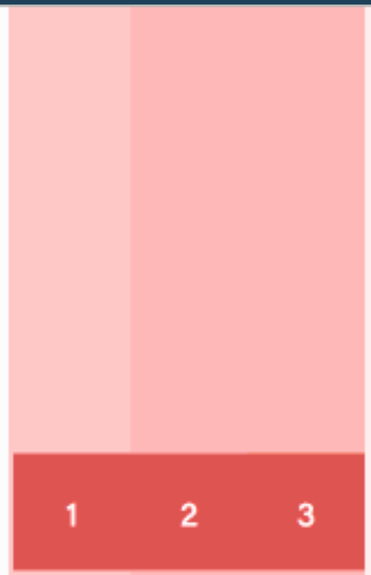
css-tricks.com/fun-viewport-units/

Flexbox

Flexbox

```
/* Default values are skipped */
{
  display: flex;
  align-content: flex-start;
}
```

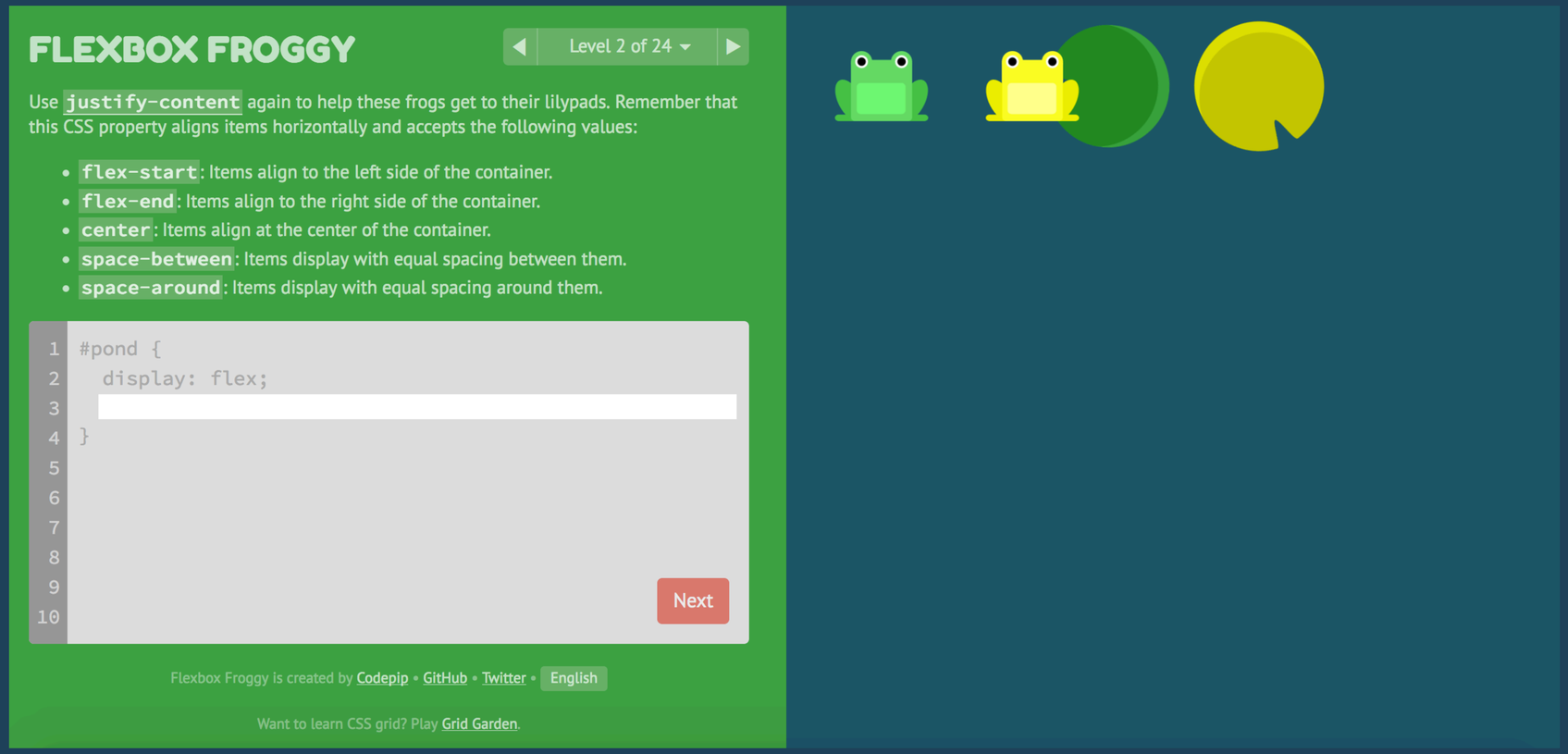
flex-direction	row column row-reverse column-reverse
flex-wrap	nowrap wrap wrap-reverse
justify-content	flex-start flex-end center space-between space-around
align-items	flex-start flex-end center baseline stretch
align-content	flex-start flex-end center space-between space-around stretch



lexbox.buildwithreact.com/#

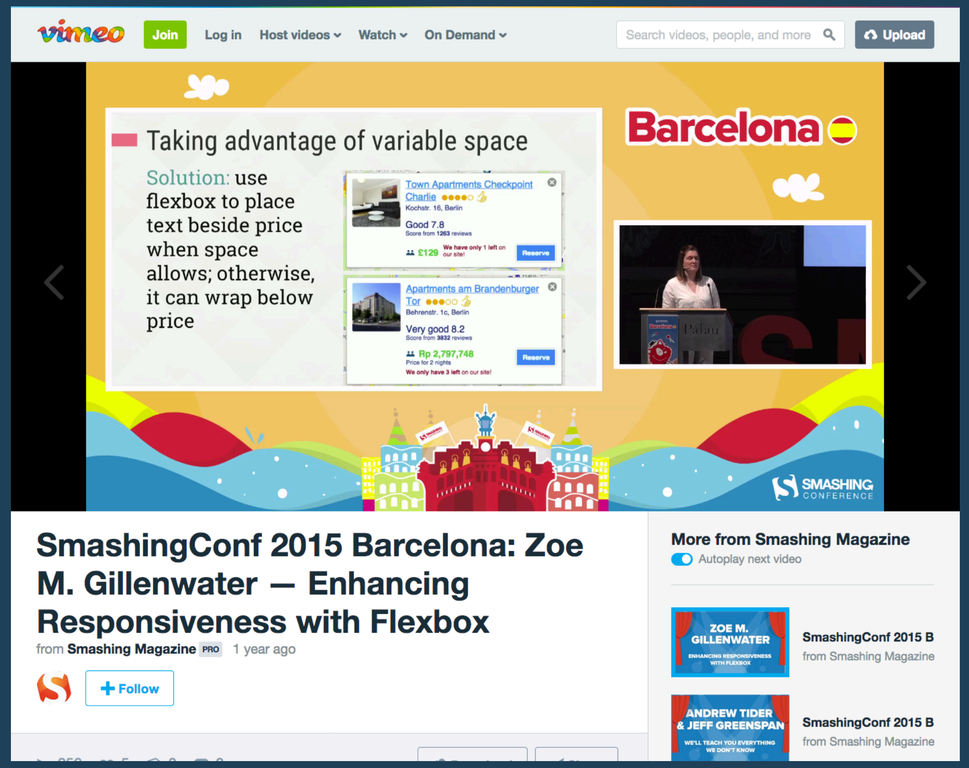
http://flexbox.buildwithreact.com/

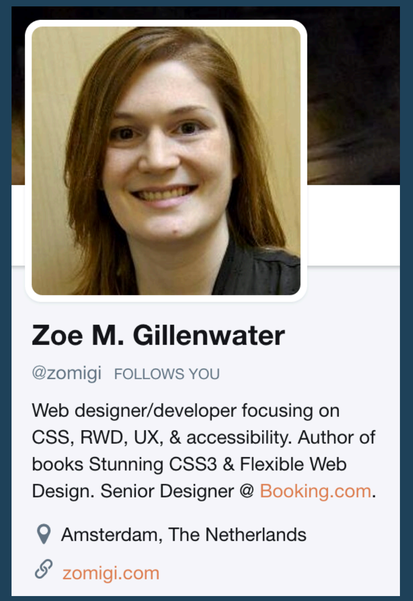
Flexbox



flexboxfroggy.com/

Flexbox





vimeo.com/145055822

Flexbox

Flexible Box Layout Module - CR

Method of positioning elements in horizontal or vertical stacks. Support includes the support for the all properties prefixed with `flex` as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Global	83.99% + 13.6% =	97.59%
unprefixed:	83.5% + 3.41% =	86.91%

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						
			57			9.2		4.4	
	14	52	58			10.2		4.4.4	
4 11	15	53	59	10.1	45	10.3	all	56	59
	16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

CSS Grid

Grid

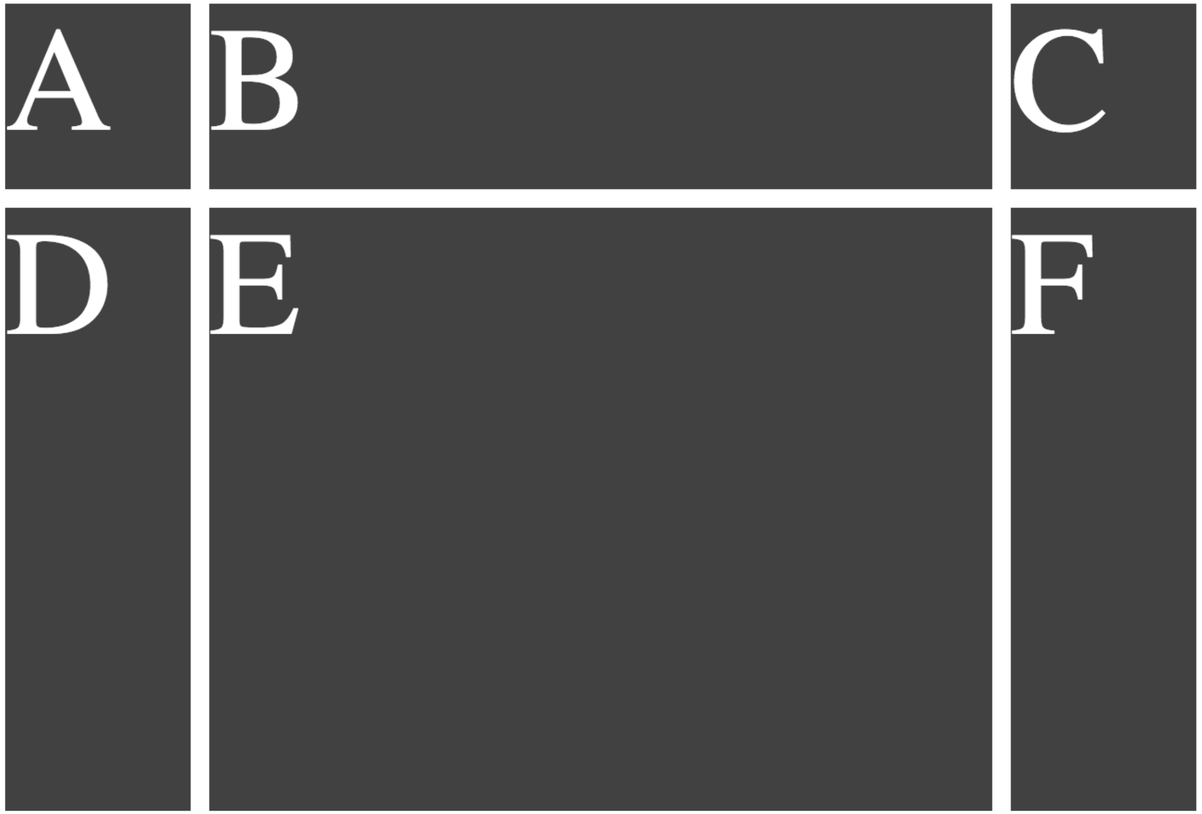
HTML

```html
<div class="wrapper">
  <div class="box a">A</div>
  <div class="box b">B</div>
  <div class="box c">C</div>
  <div class="box d">D</div>
  <div class="box e">E</div>
```

CSS

```css
.wrapper {
  display: grid;
  grid-template-columns: 100px 1fr 100px;
  grid-template-rows: 100px calc(100vh - 190px);
  grid-gap: 10px;
  background-color: #fff;
  color: #444;
}
.box {
  background-color: #444;
  color: #fff;
}
```

JS

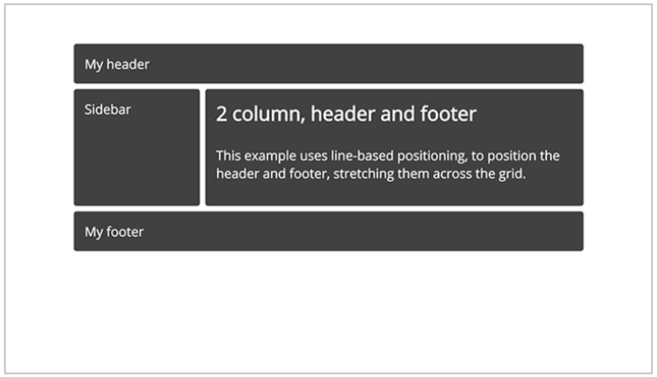


developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout

Grid

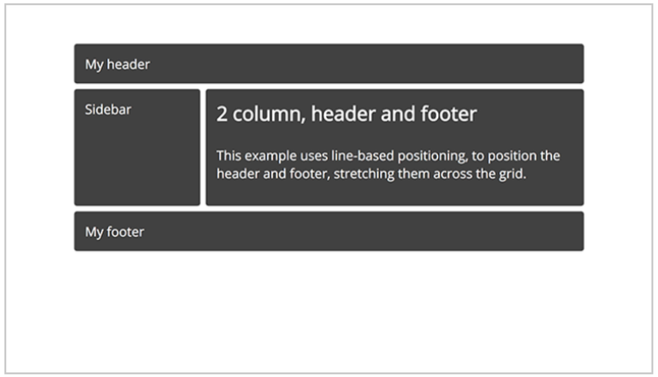
Grab & Go Patterns

A collection of built patterns to use as starting points, complete with fallbacks. Remember that you do not need to use these for full page layout, they could as easily be a small component in a design.



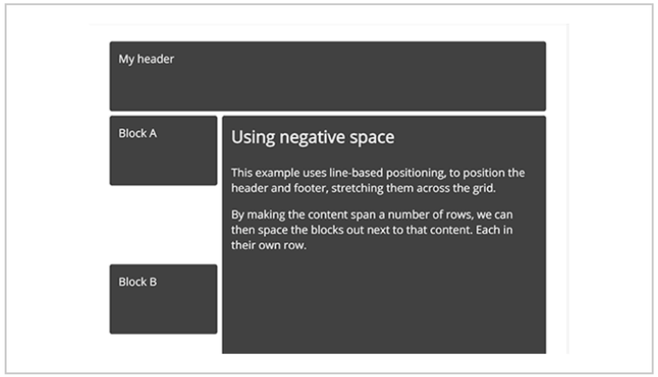
Header, 2 col, footer

A flexible, two column layout with a header and footer.



Header, 2 col, footer

A responsive, two column layout with a header and footer.



Header, 2 col, footer

A responsive, two column layout with a header and footer. Using rows to create space

gridbyexample.com

Grid

Learn Grid Layout Video Series

Learn by watching! In addition to all of the example code on this site, I'm creating a series of short vid
key features.

Defining a Grid
In this video we look at how to define a grid on a parent element.

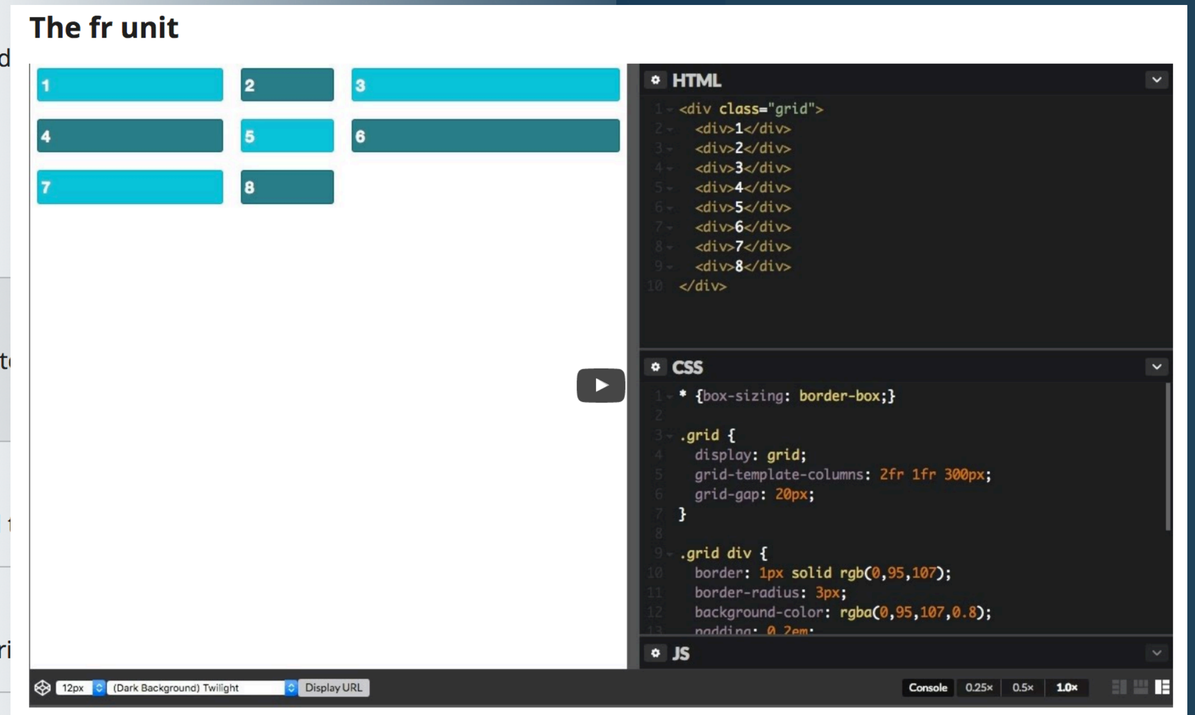
The fr unit
In this video we look at a new unit created for CSS Grid Layout - the fr unit. This unit helps us to creat
fixed width tracks in them.

Repeat Notation
In this video we take a look at repeat notation for grid tracks. I demonstrate how repeat can be used

Introducing minmax()
In this video we take a look at minmax() a method of specifying a minimum and maximum size for gri

Introducing grid auto-placement and order
In this video we take a look at the basics of how grid places child elements if we haven't specified any positioning. I also demonstrate how the order

gridbyexample.com/video/series-the-fr-unit/



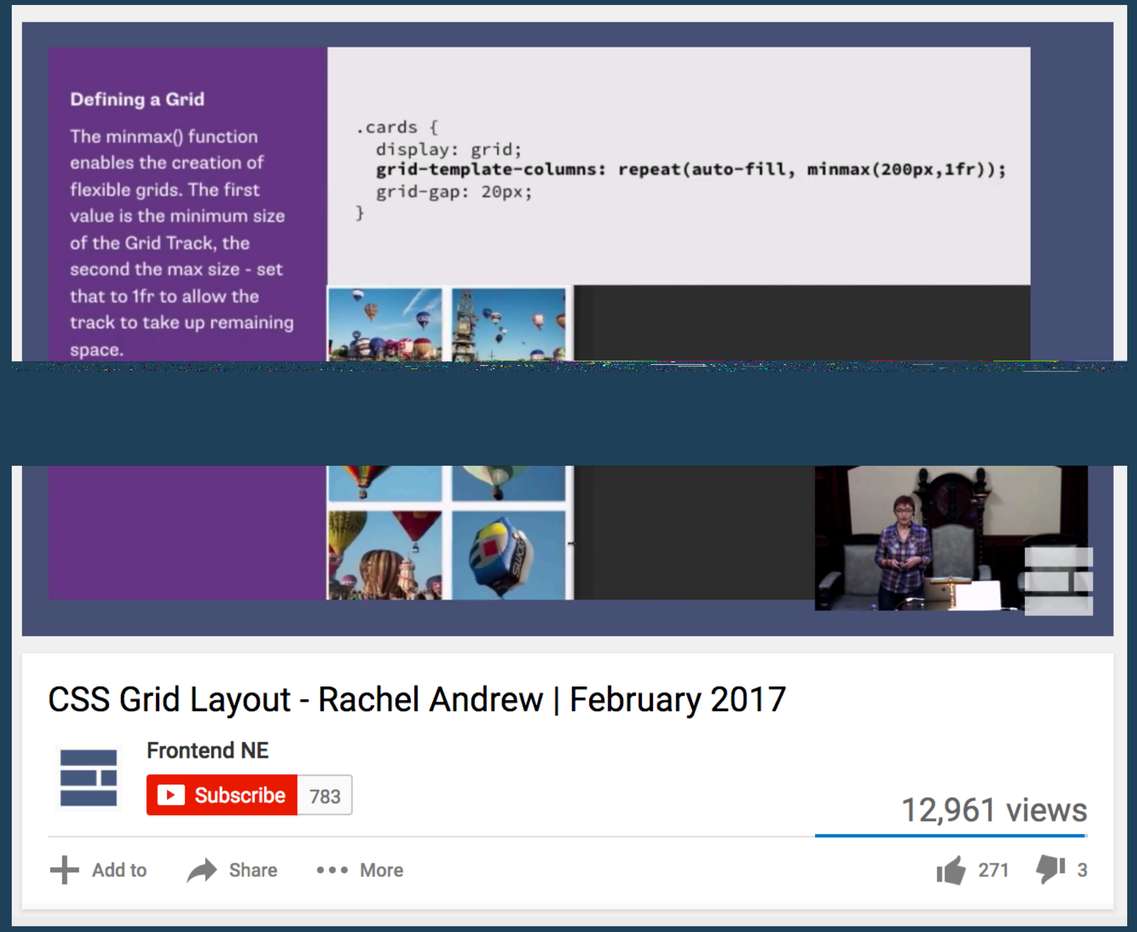
gridbyexample.com

Grid

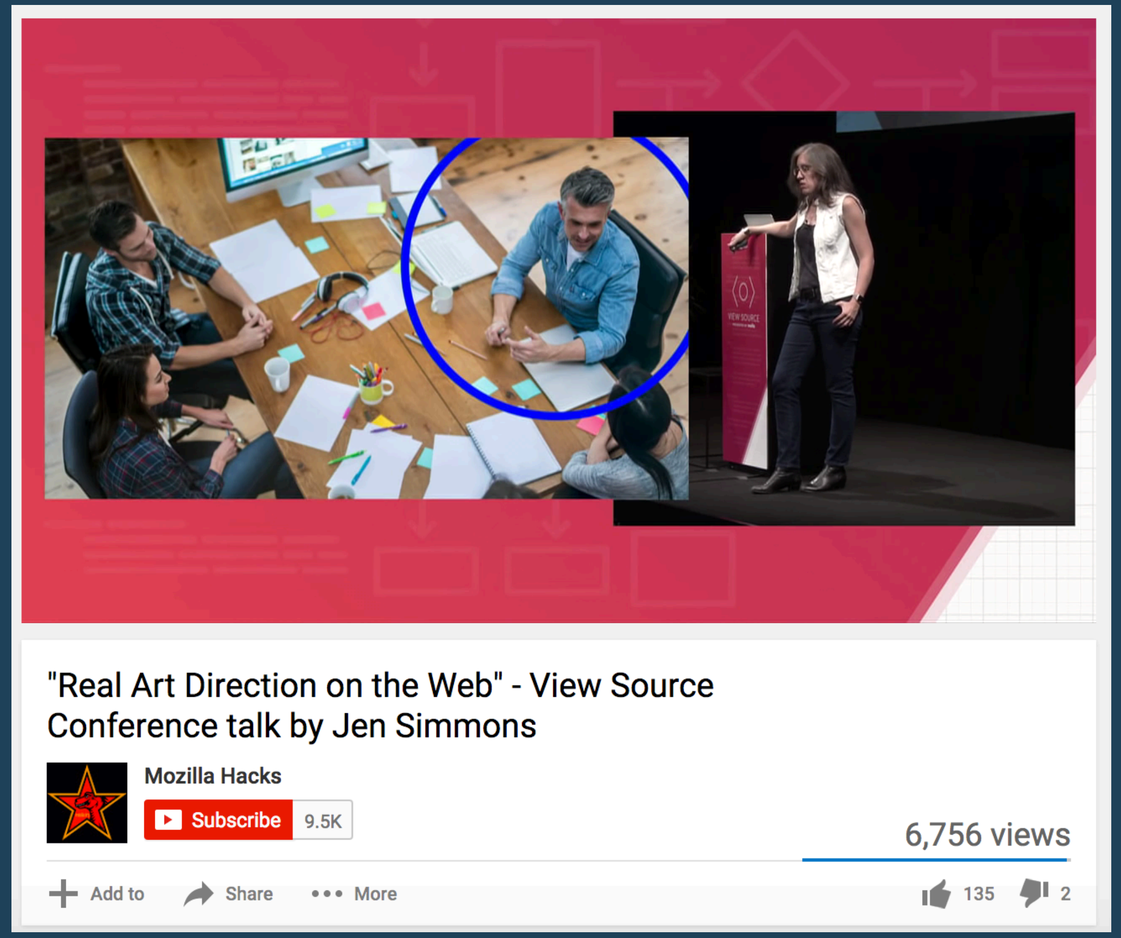


cssgridgarden.com

Grid



youtube.com/watch?v=N5Lt1SLqBmQs



youtube.com/watch?v=5Z7lSSMwRgo

Grid

CSS Grid Layout - CR

Global 63.92% + 5.39% = 69.3%

unprefixed: 63.92%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			1 49						
			1 56						
			4 57			9.2		4.4	
	2 14	4 52	58			10.2		4.4.4	
2 11	2 15	4 53	59	10.1	45	10.3	all	56	59
	2 16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

Why is there a flexbox and a grid?

Grid and flexbox

Flexbox was designed for layout in one dimension, layout in a row or a column.

Grid was designed for two-dimensional layout, layout in rows and columns at the same time.
Grid can and should use Flexbox in its cells.

developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Relationship_of_Grid_Layout

CSS Custom properties

(variables)

Custom Properties (*variables*)

```css
:root {
  --first-color: #488cff;
  --second-color: #ffff8c;
}

#firstParagraph {
  background-color: var(--first-color);
  color: var(--second-color);
}

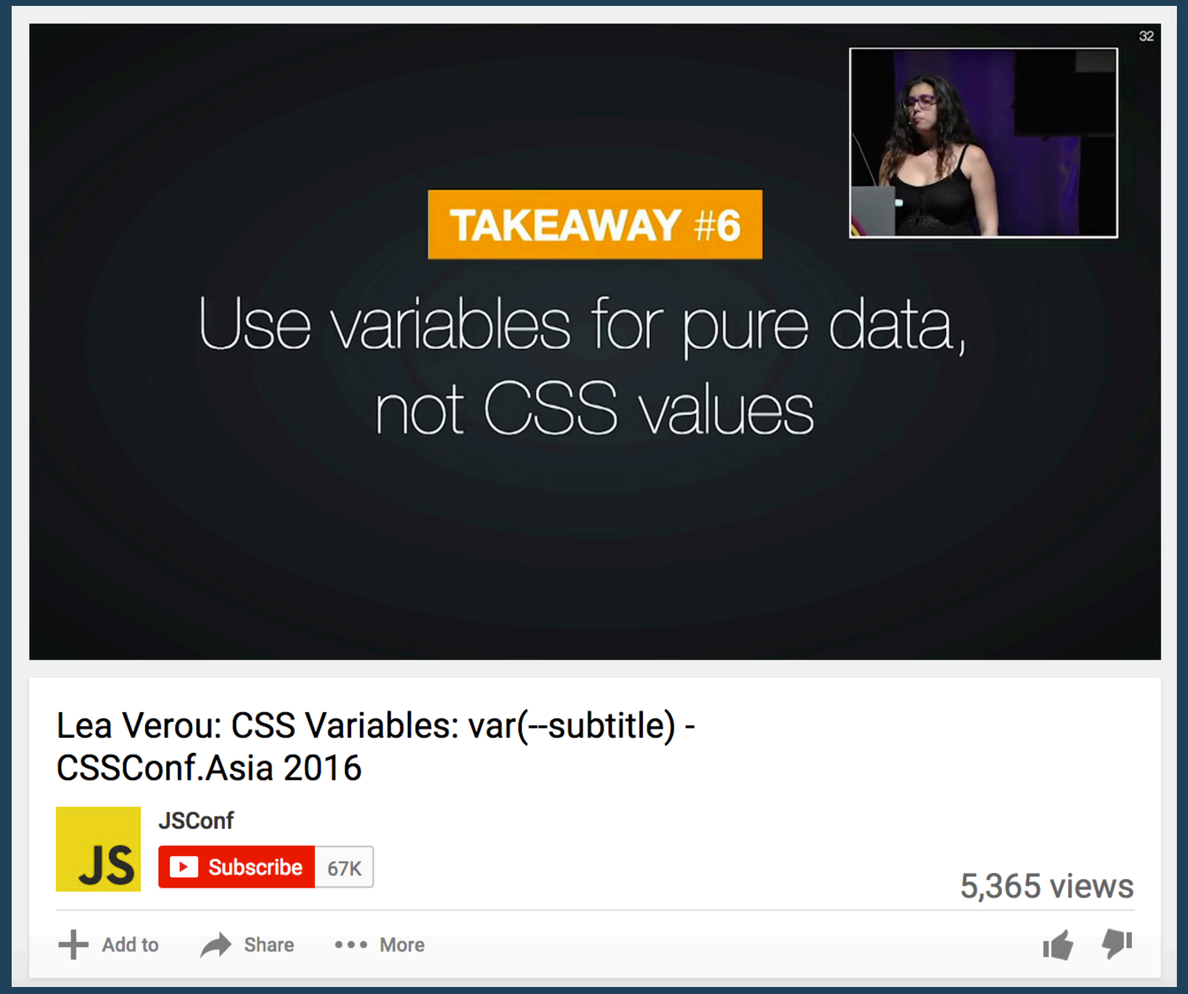
#secondParagraph {
  background-color: var(--second-color);
  color: var(--first-color);
}

#container {
  --first-color: #48ff32;
}
```

```css
.foo {
  --widthA: 100px;
  --widthB: calc(var(--widthA) / 2);
  --widthC: calc(var(--widthB) / 2);
  width: var(--widthC);
}
```

developer.mozilla.org/en-US/docs/Web/CSS/--*

Custom Properties (*variables*)



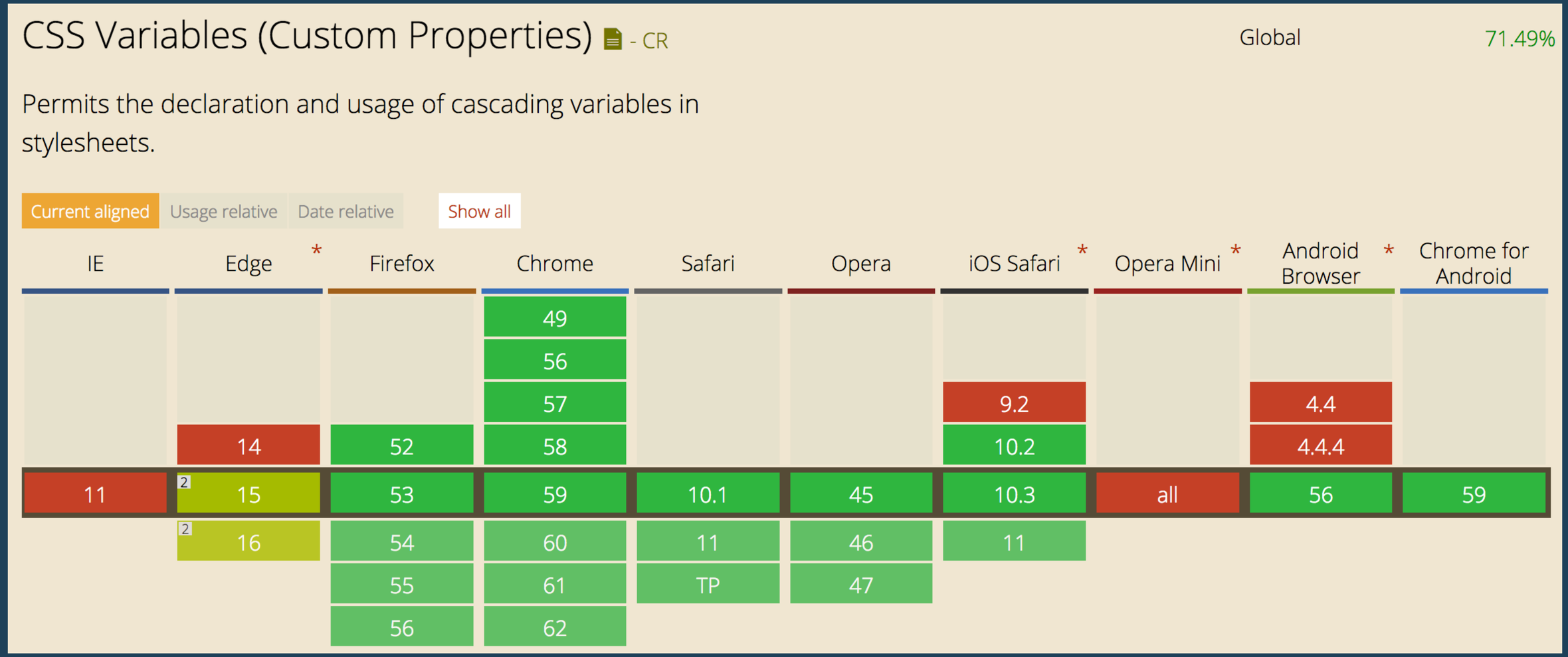

```
:root {
    --gutter: 2px;
}

section {
    margin: var(--gutter);
}

@media (min-width: 1000px) {
    :root {
        --gutter: 10px;
    }
}
```

youtube.com/watch?v=kZOJCVvyF-4

Custom Properties (*variables*)



CSS Feature Queries

CSS Feature Queries

```
@supports (animation-name: test) {
    … /* applied when animations are supported unprefixed */
    @keyframes {
      …
    }
}
```

https://developer.mozilla.org/en-US/docs/Web/CSS/@supports

CSS Feature Queries

CSS Feature Queries - CR

Global	92.23%
U.K.	89.55%

CSS Feature Queries allow authors to condition rules based on whether particular property declarations are supported in CSS using the @supports at rule.

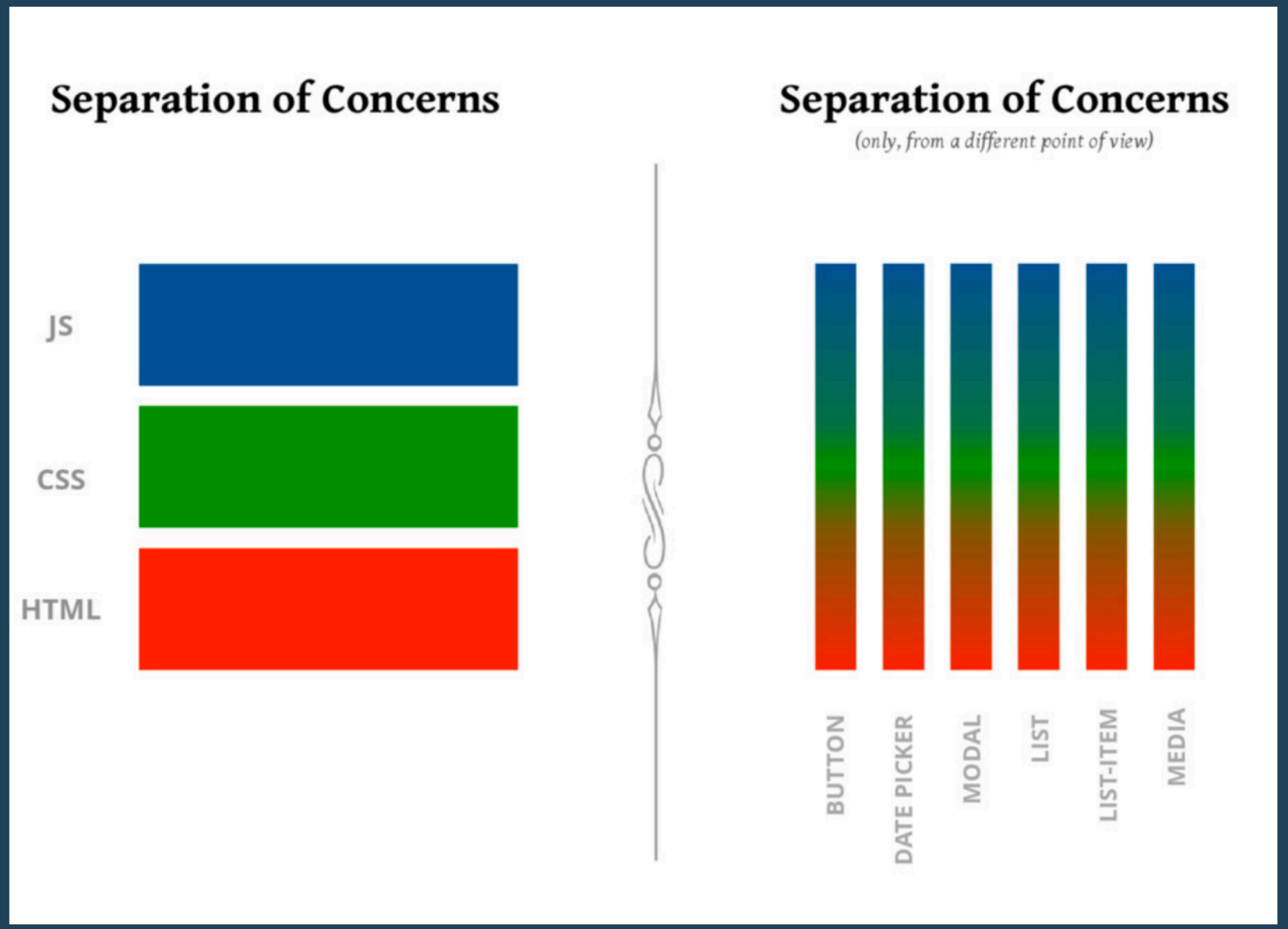
Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56						
			57			9.2		4.4	
	14	52	58			10.2		4.4.4	
11	15	53	59	10.1	45	10.3	all	56	59
	16	54	60	11	46	11			
		55	61	TP	47				
		56	62						

CSS and JavaScript?

CSS and JavaScript working together is incredibly powerful and the right thing to do.

CSS can’t solve everything…



Cristiano Rastelli

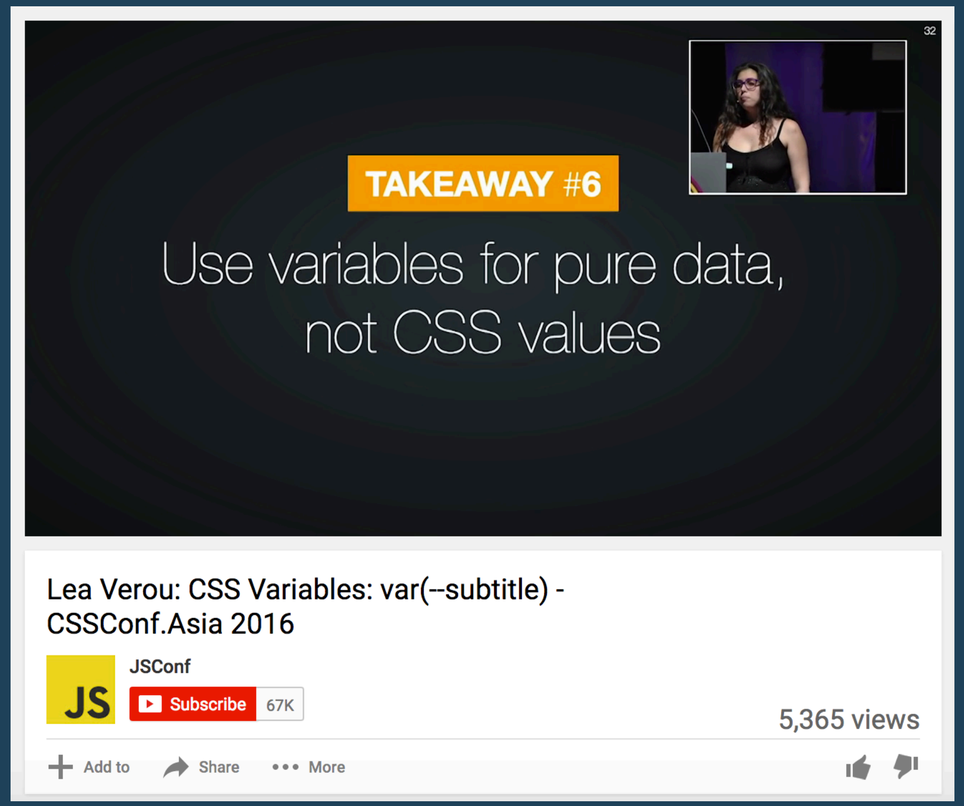
@areaweb

speakerdeck.com/didoo/let-there-be-peace-on-css

CSS and JavaScript working together in a non-module world…

- Use CSS when you can using the things you saw here
- If you need to communicate with CSS, consider changing Custom Properties
- If that’s not an option apply classes to parent elements using classList.
- As a very last resort, you can alter the style directly

CSS and JavaScript working together in a non-module world…



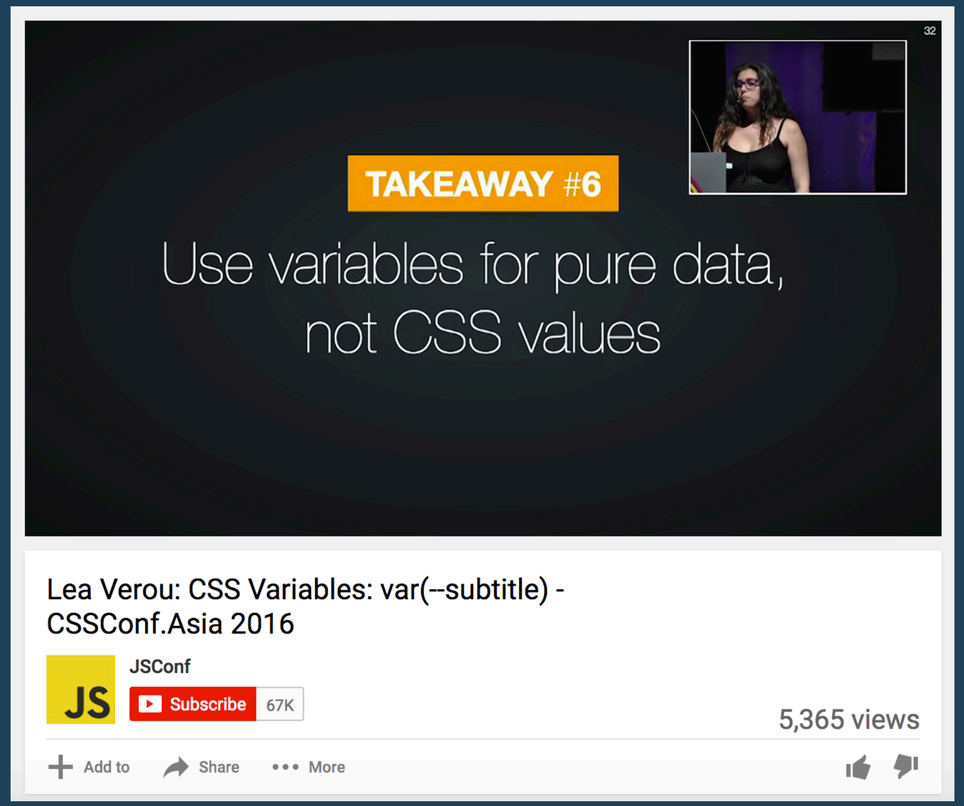
youtube.com/watch?v=kZOJCVvyF-4

```
// Get variable from inline style
element.style.getPropertyValue("--foo");

// Get variable from wherever
getComputedStyle(element).getPropertyValue("--foo");

// Set variable on inline style
element.style.setProperty("--foo", 38 + 4);
```

CSS and JavaScript working together in a non-module world…



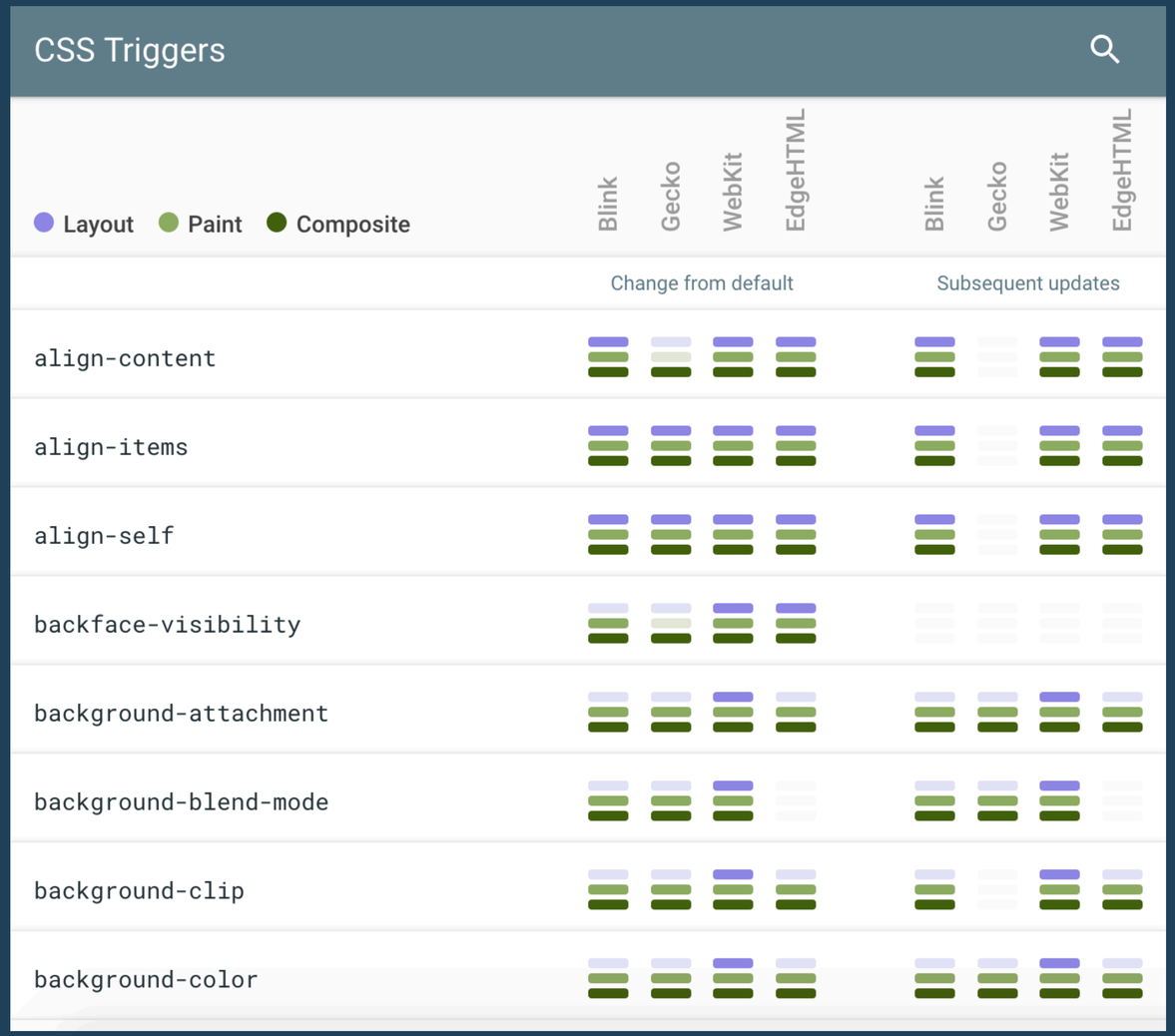
youtube.com/watch?v=kZOJCVvyF-4

```
var root = document.documentElement;

document.addEventListener("mousemove", evt => {
    let x = evt.clientX / innerWidth;
    let y = evt.clientY / innerHeight;

    root.style.setProperty("--mouse-x", x);
    root.style.setProperty("--mouse-y", y);
});
```

Every style change has consequences…



Paul Lewis @aerotwist

Das Surma @DasSurma

csstriggers.co
m


CSS is much
more reliable than
it used to be…

> Users over authors over implementors over specifiers over theoretical purity…

https://www.w3.org/TR/html-design-principles/

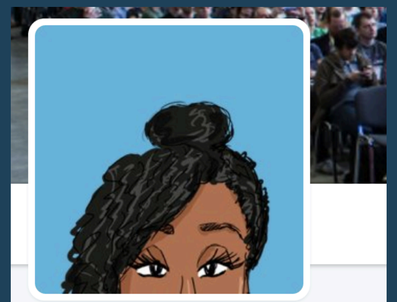
Our users deserve interfaces that are smooth, reliable and don’t kill their batteries.

So, consider CSS a bit more…



You can be lazy and build on the work of the community…





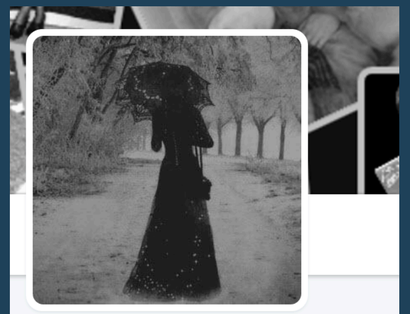
Ire Aderinokun

@ireaderinokun FOLLOWS YOU

Frontend Developer and UI Designer / Software Developer at @eyeo / Blog at bitsofco.de (@botsofcode) / Google Expert in Web Tech

Lagos, Nigeria

ireaderinokun.com



Ana Tudor 🐯

@anatudor FOLLOWS YOU

Don't follow me, I'm lost too. Mad scientist. Homeless & unemployed. Not friendly. Not a dev/ designer. Background in electronics/ Maths.

thebabydino.github.io



Rachel Andrew

@rachelandrew FOLLOWS YOU

Web developer, writer, speaker, runner. Half of the company behind @grabaperch.

Bristol, UK

rachelandrew.co.uk



Chris Coyier

@chriscoyier FOLLOWS YOU

Writer & founder @real_css_tricks // Co-founder & designer @codepen (Check out CodePen Projects: codepen.io/pro/projects/) // Podcaster @shoptalkshow

Milwaukee, WI

chriscoyier.net



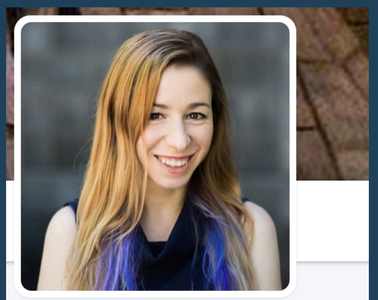
Jen Simmons

@jensimmons

Designer Advocate at Mozilla. Host & executive producer of The Web Ahead podcast. Excited about new CSS for web page layout & revolutionizing editorial design.

Brooklyn, New York

jensimmons.com



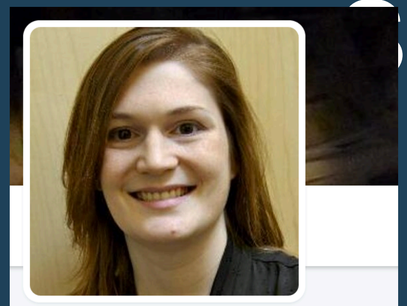
Sarah Drasner

@sarah_edo FOLLOWS YOU

Award-winning speaker. Consultant. Writer @Real_CSS_Tricks. Cofounder, webanimationworkshops.com, formerly @trulia (Zillow). codepen.io/sdras/

sarahdrasnerdesign.com

Super inspiring and highly active CSS folk!



Zoe M. Gillenwater

@zomigi FOLLOWS YOU

Web designer/developer focusing on CSS, RWD, UX, & accessibility. Author of books Stunning CSS3 & Flexible Web Design. Senior Designer @ Booking.com.

Amsterdam, The Netherlands

zomigi.com



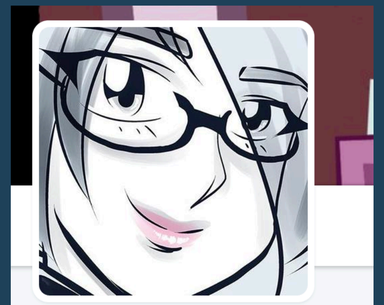
Brad Frost 🌎

@brad_frost FOLLOWS YOU

I am a web designer, speaker, writer, consultant, and musician in beautiful Pittsburgh, PA. Author of Atomic Design. #resist #raisethatflag

Pittsburgh, PA

bradfrost.com



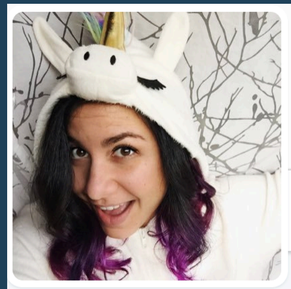
Rachel 🦄Nabors

@rachelnabors FOLLOWS YOU

Curator: webAnimationWeekly.com Founder: slack.animationatwork.com GDE. W3C. Design advocate @MSEdgeDev Tweets my own. Animating the Web forward.

Seattle, WA, USA

rachelnabors.com



Una Kravets 👩🏻‍💻

@Una FOLLOWS YOU

Future bosslady. UI Engineer @digitalocean. Community builder, tech writer & cohost @toolsday. Part-time nomad: Instagram.com/unakravets

New York

una.im



Lea Verou

@LeaVerou FOLLOWS YOU

HCI researcher @MIT_CSAIL, @CSSWG IE, @CSSSecretsBook author, Ex @W3C staff. Made @mavoweb @prismjs @dabblet. I ❤️ standards, code, design, UX, life!

Boston, MA

lea.verou.me



Sara Soueidan

@SaraSoueidan FOLLOWS YOU

Front-end developer. I write & speak about CSS, SVG, UI/X, a11y, JS. net Developer of the Year 2015. You can hire me 👉 sarasoueidan.com/hire

Lebanon

sarasoueidan.com

THANKS!

CHRIS HEILMANN

@CODEPO8

CHRISTIANHEILMANN.COM

