

CSS-in-JS

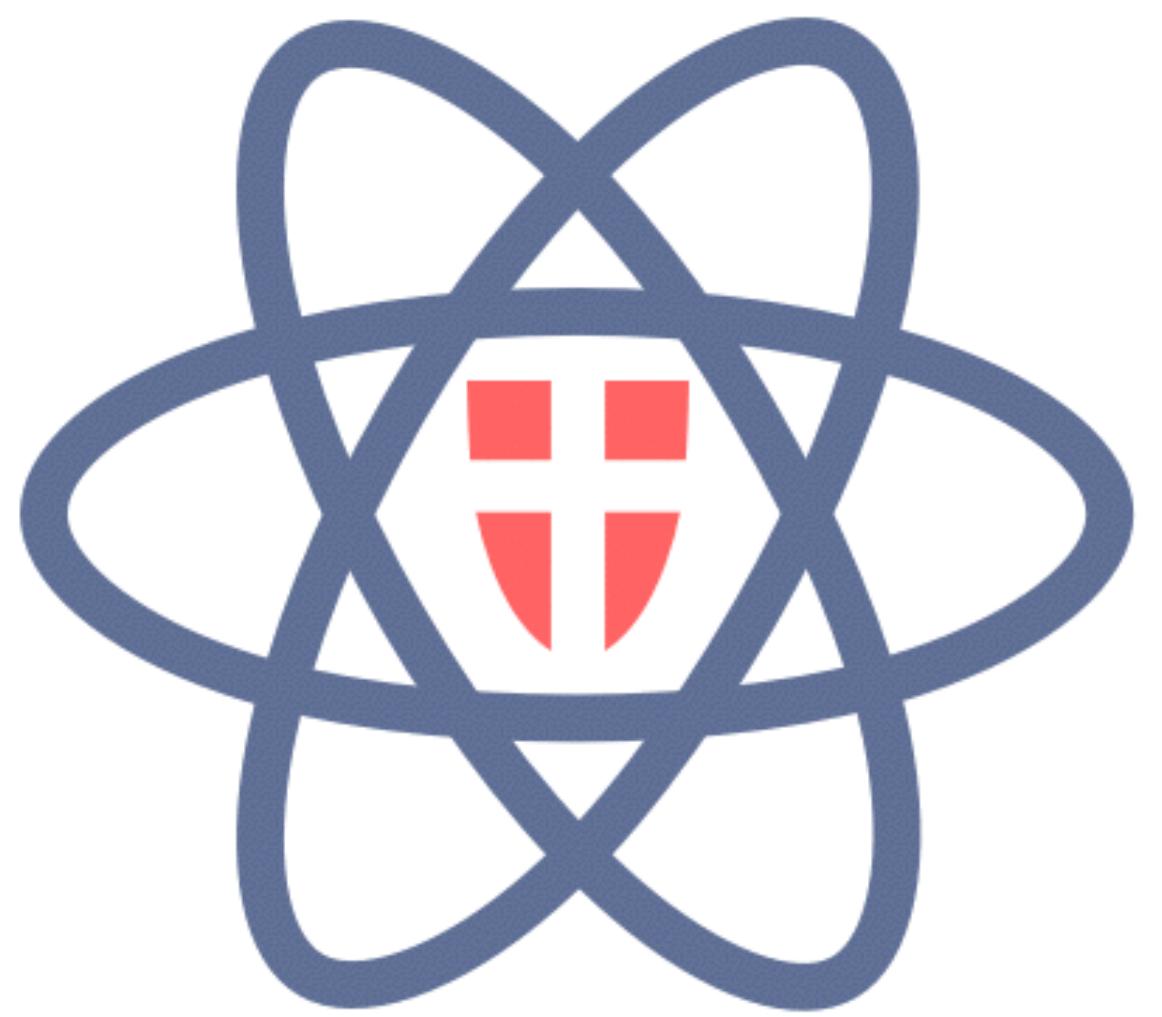
The good or the evil?



Andrey Okonetchnikov

@okonetchnikov

React



Vienna



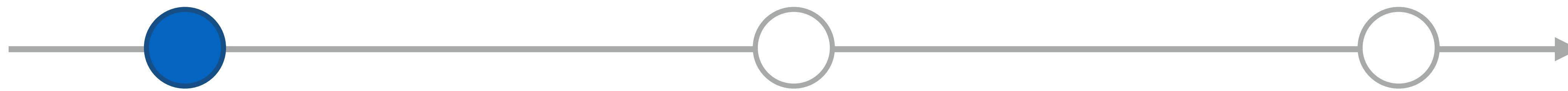
ColorSnapper

<http://colorsnapper.com>

Past

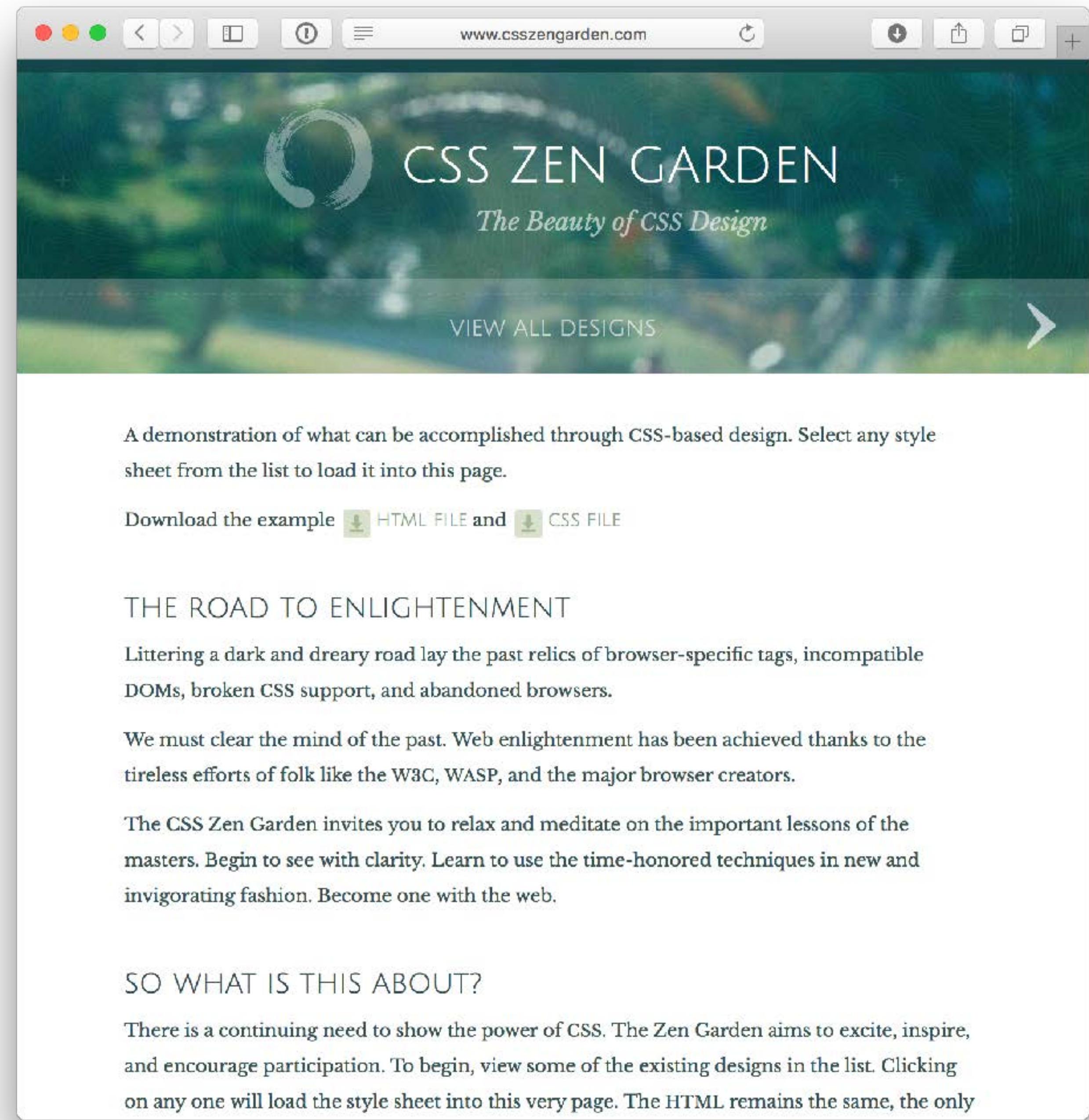
Current time

Future



A brief history of CSS

*CSS was designed for **documents***

A screenshot of a web browser displaying the CSS Zen Garden homepage. The browser window has a white header bar with standard OS X-style buttons (red, yellow, green) and a search bar containing the URL 'www.csszengarden.com'. The main content area features a large, blurred background image of a garden with green foliage and a path. Overlaid on this image is a white circular logo on the left and the text 'CSS ZEN GARDEN' in large, bold, white capital letters, with the subtitle 'The Beauty of CSS Design' in a smaller, italicized, white serif font below it. A white button with the text 'VIEW ALL DESIGNS' and a right-pointing arrow is positioned in the lower right corner of the main area. Below the main image, there is a white text area containing descriptive text and download links. The text area includes a heading 'THE ROAD TO ENLIGHTENMENT', a paragraph about the history of web design, a paragraph about the goals of the Zen Garden, and a heading 'SO WHAT IS THIS ABOUT?' followed by a paragraph explaining the purpose of the site.

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [HTML FILE](#) and [CSS FILE](#)

THE ROAD TO ENLIGHTENMENT

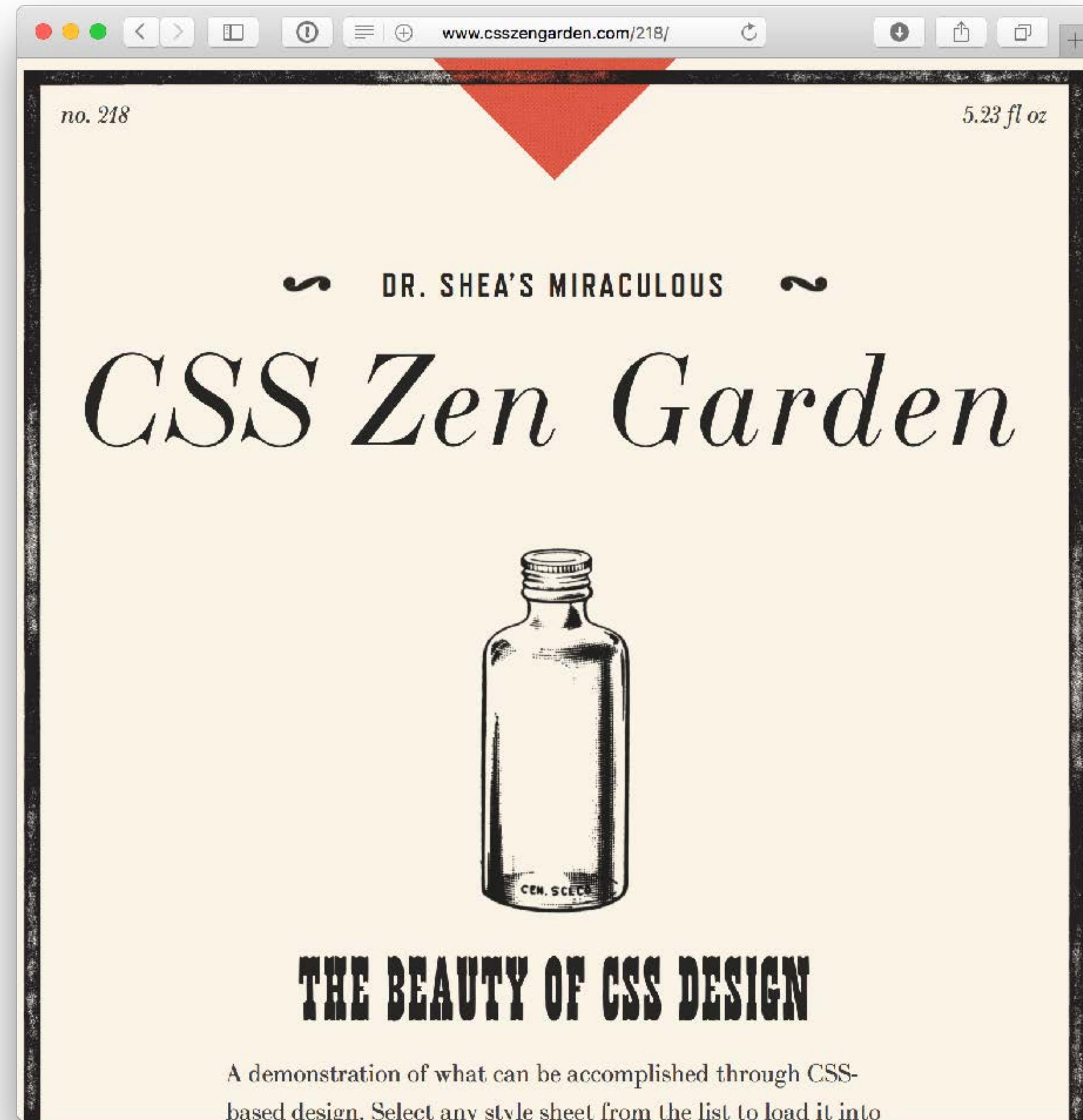
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WASP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only





www.csszengarden.com/221/

The Beauty of CSS Design



A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

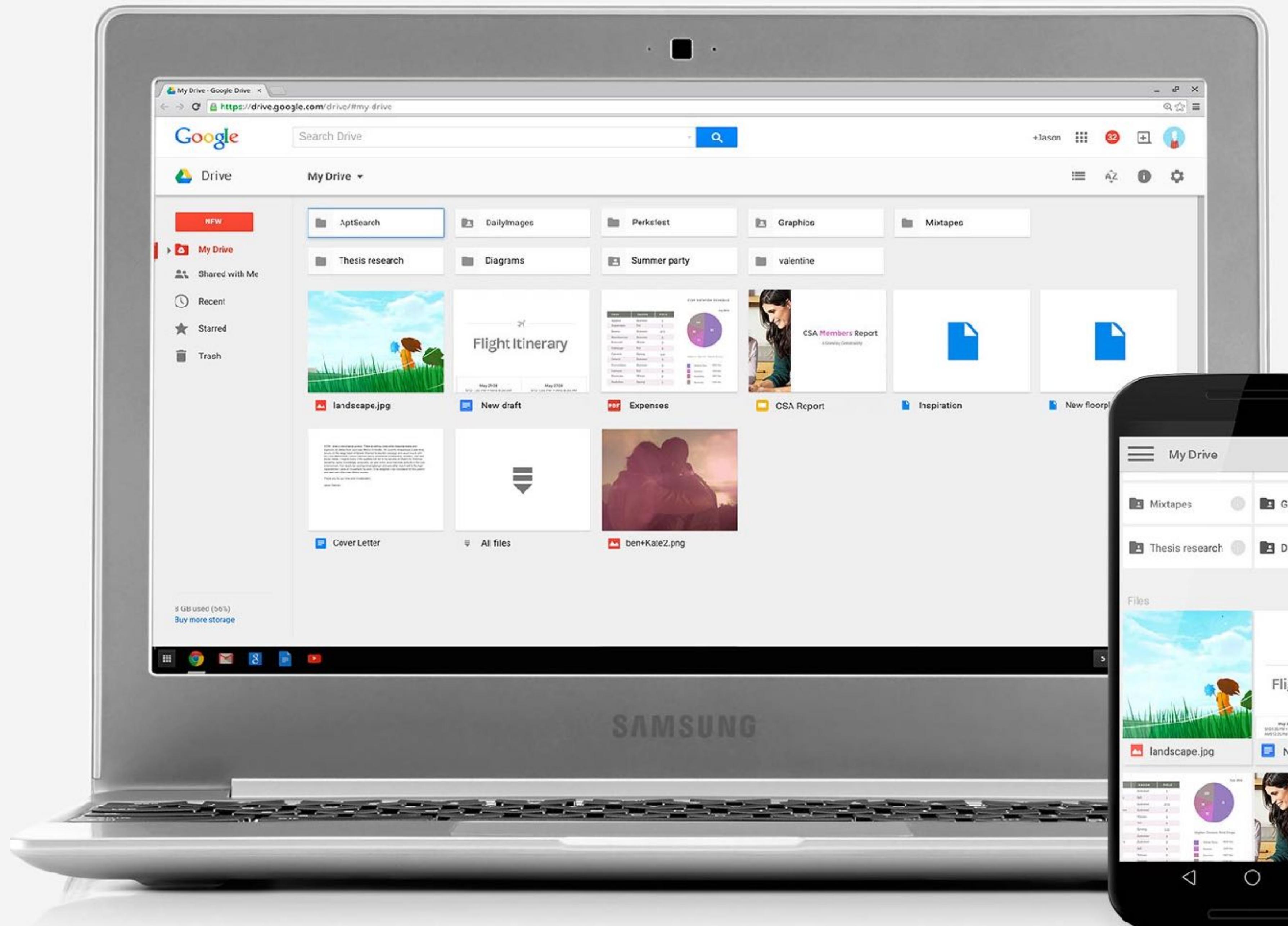
Download the example [html file](#) and [css file](#)





*How many of you have ever re-designed a
web-site with CSS changes only?*

*CSS was designed for **documents**,
not for **web-applications***





More ▾

1-50 of 5,954

Primary**Social 3 new**
Google+, YouTube, Em...**Promotions 2 new**
Google Offers, Zagat**Updates 2 new**
Shoehop, Blitz Air

- ★ Benedict Hieger You won't believe who I found! - I was walking down Stockton when
- ★ Michael Gurman My new apartment - what do you think? Pics attached
- ★ Jenny Kang Should we go to this concert? - Hey, remember that b
- ★ Max Stein I won't be able to walk your dog this week. - Hey Jess
- ★ Emily Million Hey lady! - New rug for our place? Check out this rug
- ★ Jonathan Pelleg Surprise Party! - Yo, was thinking a surprise party for
- ★ Ronnie Allman I'm in the news today! - Hey, they featured my project!
- ★ Benedict Hieger You won't believe who I found! - I was walking down S
- ★ Michael Gurman My new apartment - what do you think? Pics attached
- ★ Jenny Kang Should we go to this concert? - Hey, remember that b
- ★ Max Stein I won't be able to walk your dog this week. - Hey Jess
- ★ Emily Million Hey lady! - New rug for our place? Check out this rug
- ★ Jonathan Pelleg Surprise Party! - Yo, was thinking a surprise party for
- ★ Emily Million Hey lady! - New rug for our place? Check out this rug
- ★ Jonathan Pelleg Surprise Party! - Yo, was thinking a surprise party for
- ★ Ronnie Allman I'm in the news today! - Hey, they featured my project!

Follow up

Evan Keller

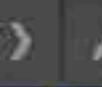
Follow up

I was great meeting up with you, thanks for all the helpful advice, we should try to next official for next month sometime

Best!

Send



W Page: Home  Publish 

featured image

 latest popular for you genres 

latest reviews

Background image album artist 
Lorem ipsum dolor sit amet, consectetur adipiscing

Background image album artist 
Lorem ipsum dolor sit amet, consectetur adipiscing

Background image album artist 
Lorem ipsum dolor sit amet, consectetur adipiscing

Div Block Selector: States color fill 

1 on this page

Layout Advanced

Display Setting  ABC ABC  

Width: 100% Height: 100% Min: 0px Max: None

Min: 0px Max: None

Overflow:    

Position: Auto 

Body  z-index: 1

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

Body      

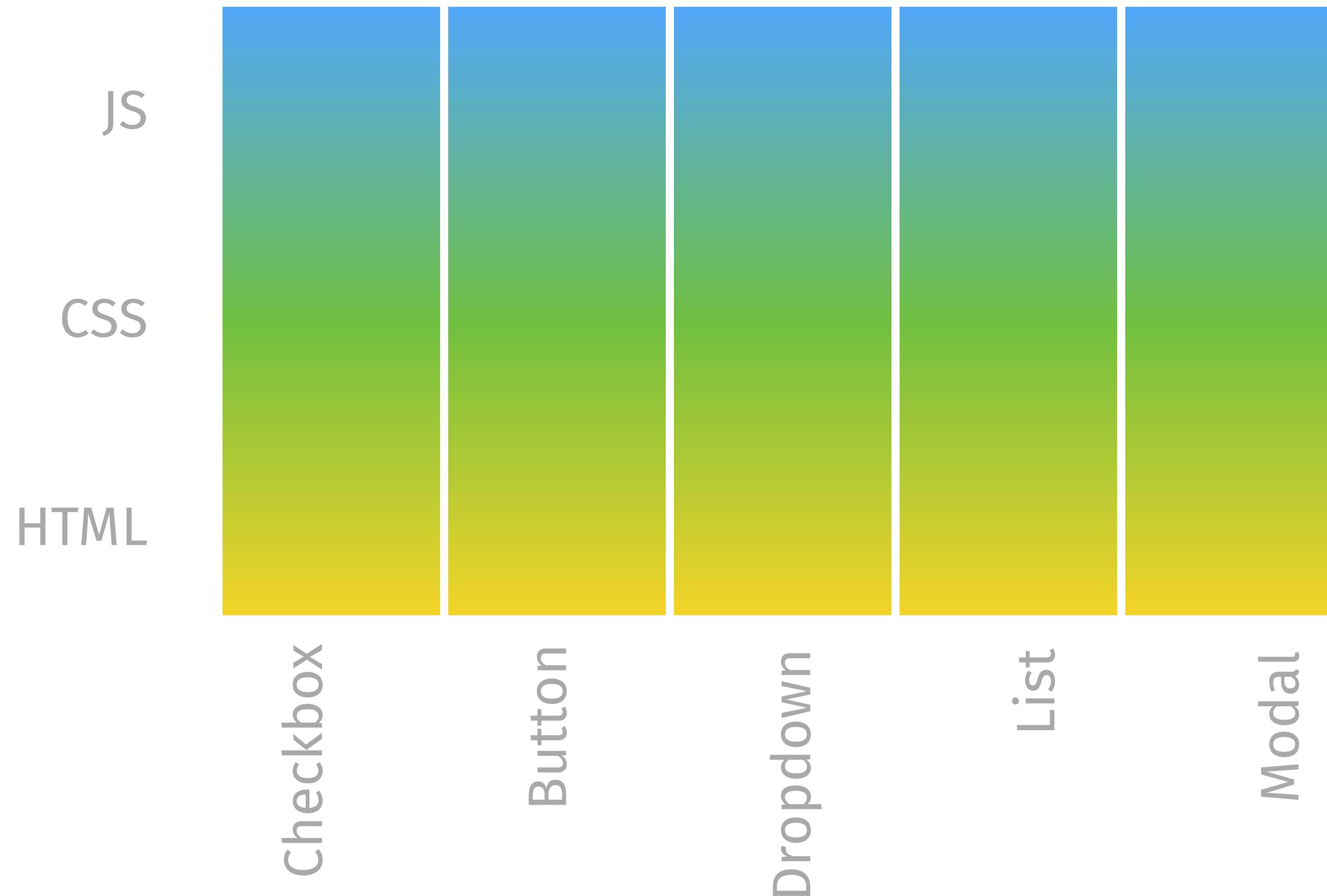
Body   <img alt="Edit

Best practices

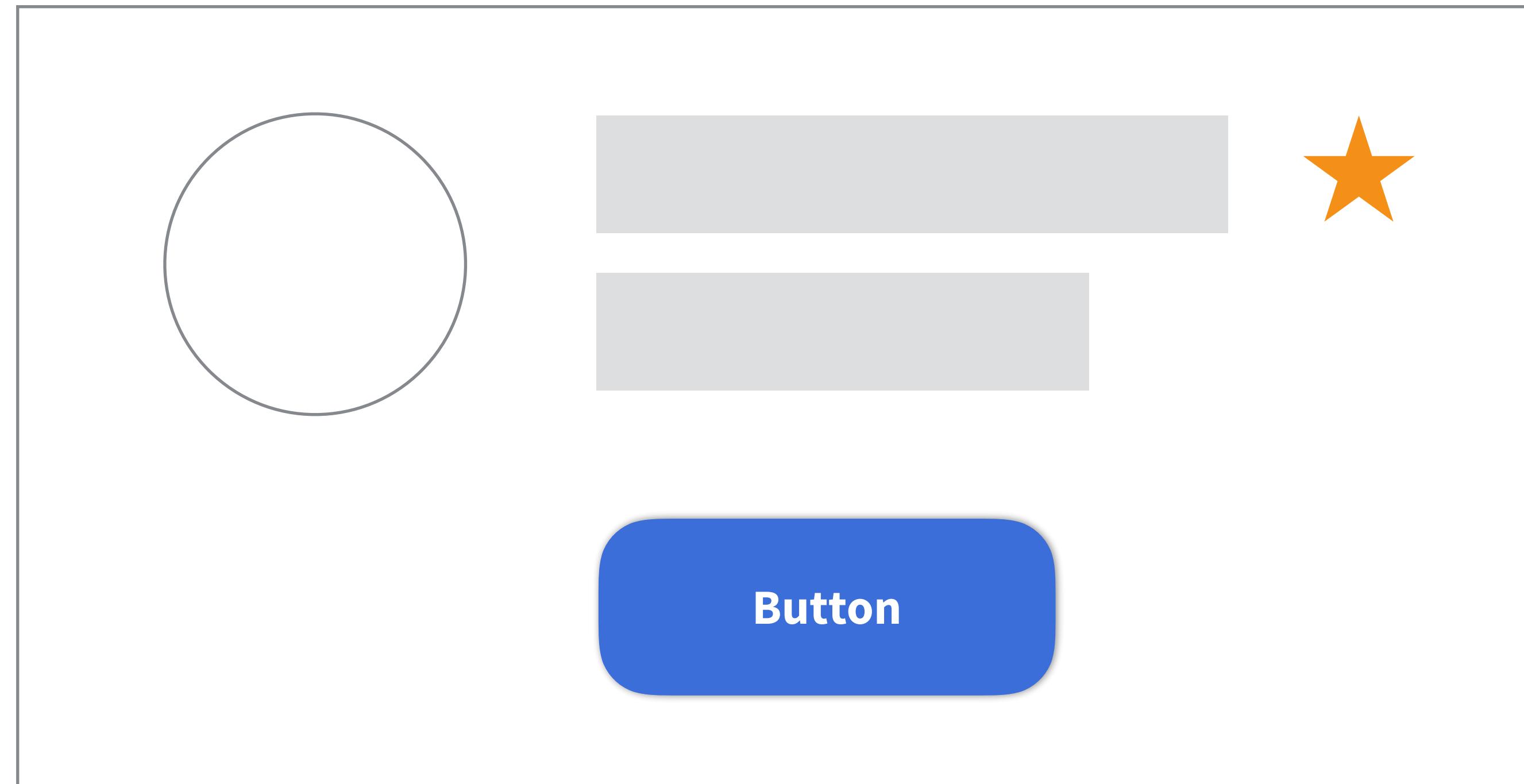
Separation of concerns



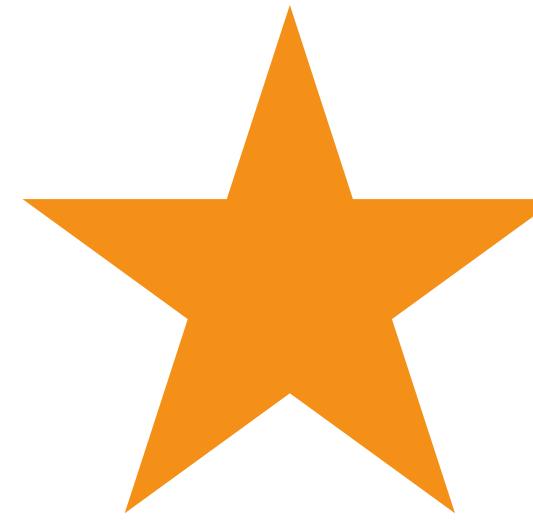
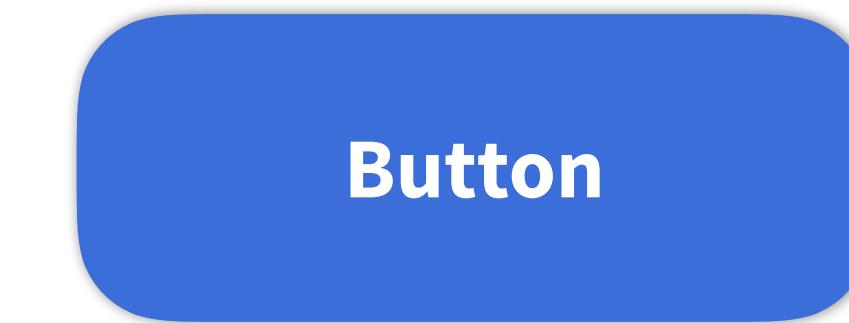
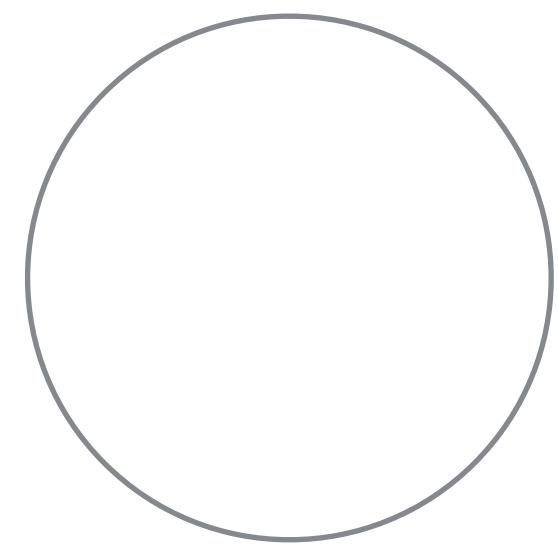
Separation of concerns

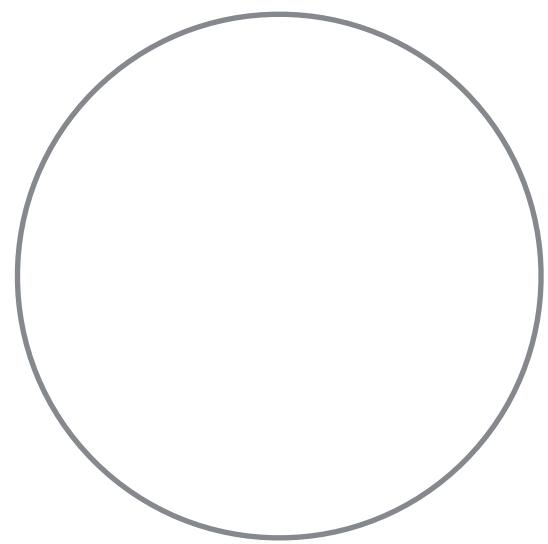


UI Components

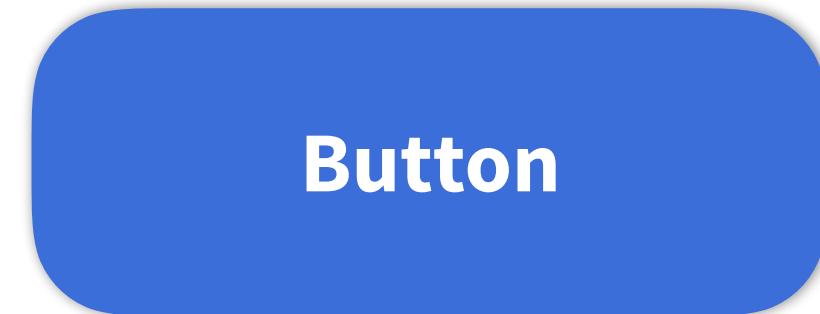


UI Components





- Logic
- HTML
- Styles



- Logic
- HTML
- Styles

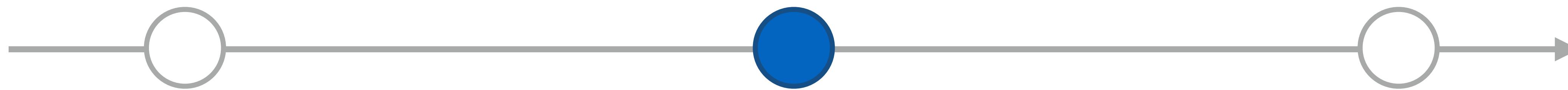


- Logic
- HTML
- Styles

Past

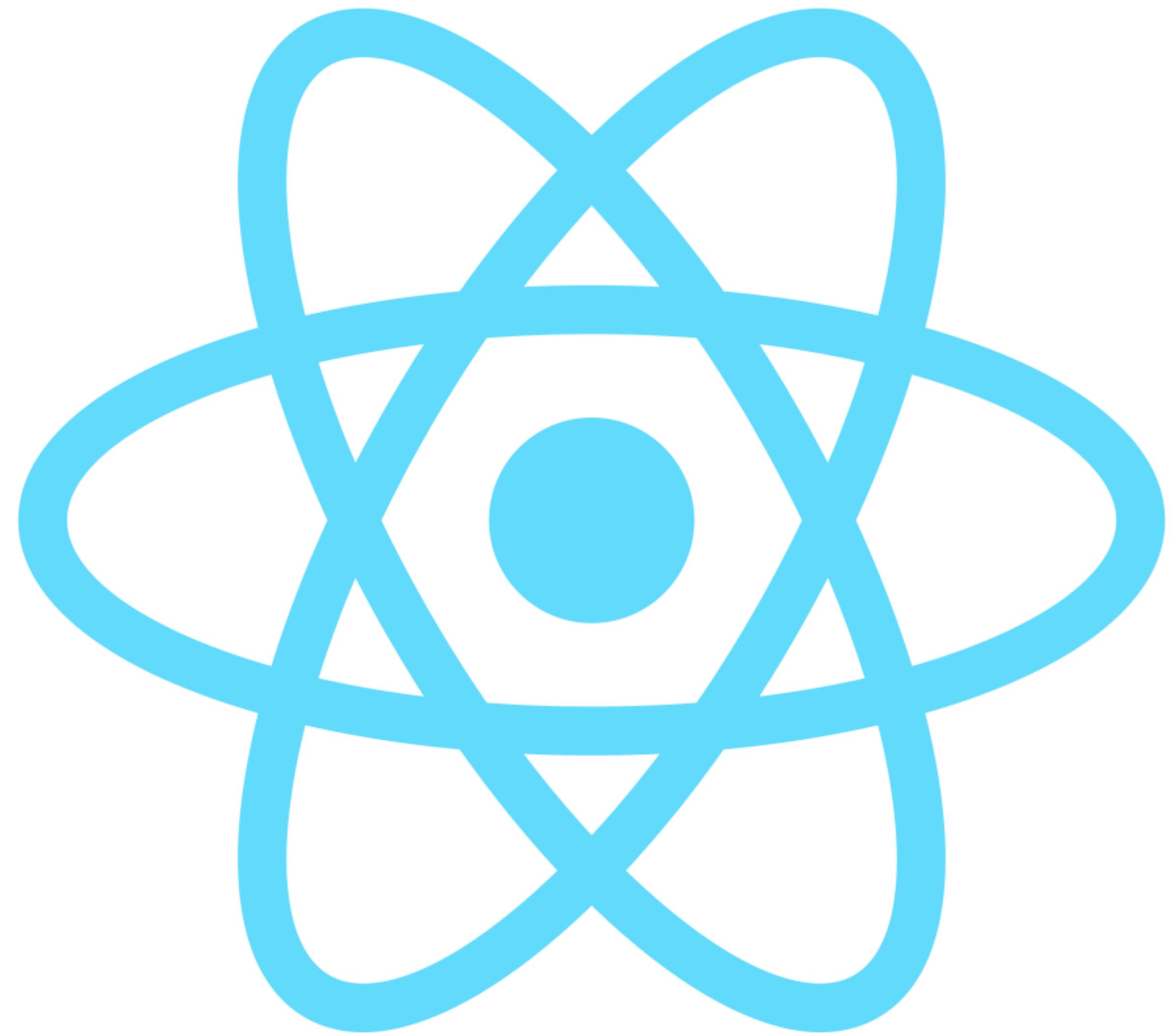
Current time

Future

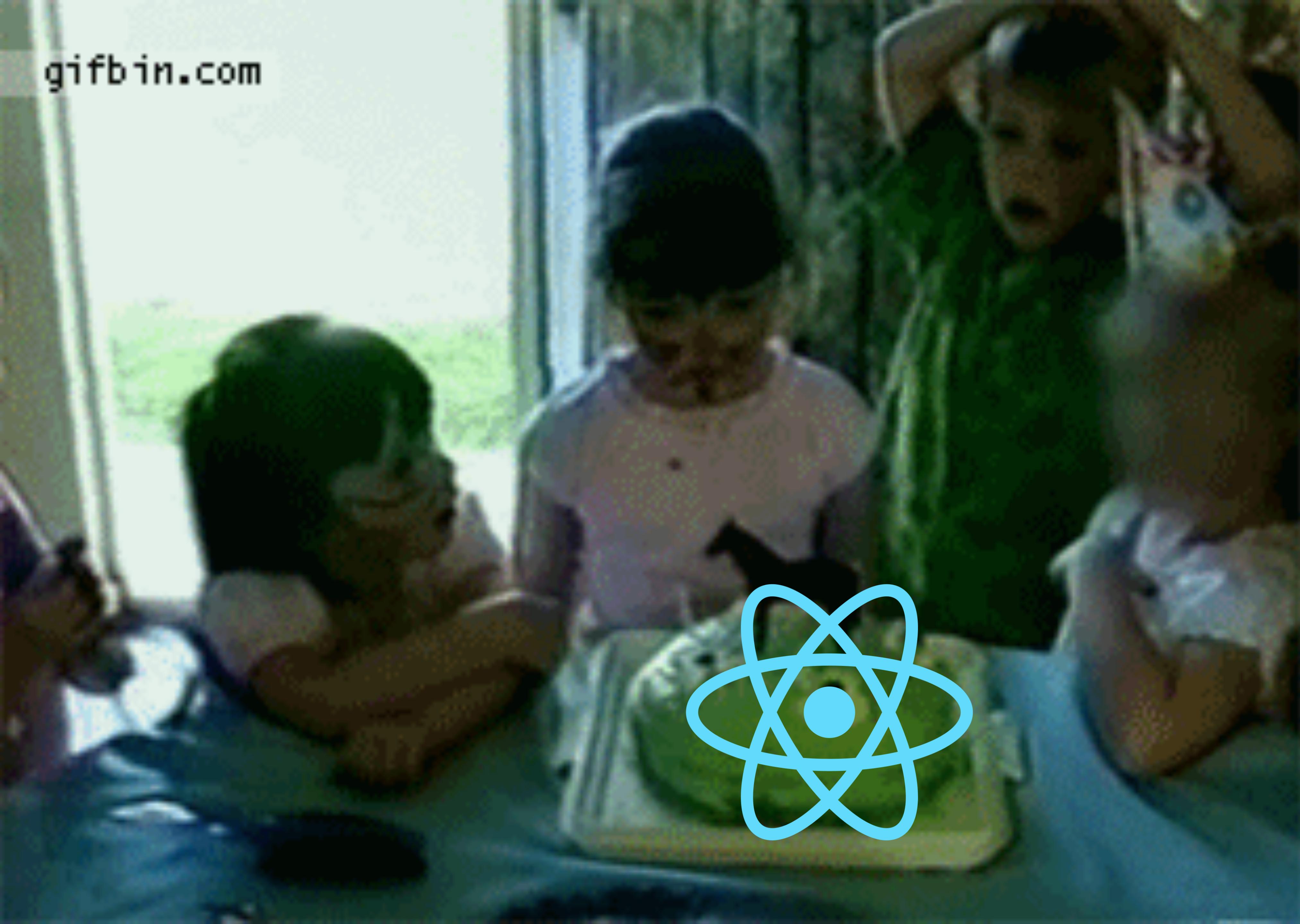


HTML-in-JS

a.k.a. JSX



gifbin.com



react is a terrible idea

Get Device Privacy Tips! | [Twitter](#) | [≡](#)

Web Images Videos News

Austria ▾ Safe Search: Strict ▾ Any Time ▾ [Send feedback](#)

Panda Strike: React Is A Terrible Idea
React Is A Terrible Idea. Mar 11, 2015 Dan Yoder. ... Even if React was a good design (it's not), that's still severely limiting the potential audience.
<https://www.pandastrike.com/posts/20150311-react-bad-idea>

React Is A Terrible Idea • r/programming - reddit
React Is A Terrible Idea ... React is a fine idea. ... are pathologically incapable of acknowledging that putting template markup in JavaScript is a terrible idea.
https://www.reddit.com/r/programming/comments/2ysm8t/react_is_a_...

React Is a Terrible Idea | Hacker News
[flagged] React Is a Terrible Idea (pandastrike.com) 61 points by Zelphyr 159 days ago ... thatswrong0 159 days ago. To summarize: > React, Bad; Web Components, Good.
<https://news.ycombinator.com/item?id=12101849>

React Is A Terrible Idea (Webcomponents are better) - reddit
reddit: the front page of ... React Is A Terrible Idea (Webcomponents are better) (pandastrike.com) ... Every new Idea is bad because that's not the way we do things.
https://www.reddit.com/r/javascript/comments/2ysm7o/react_is_a_t...

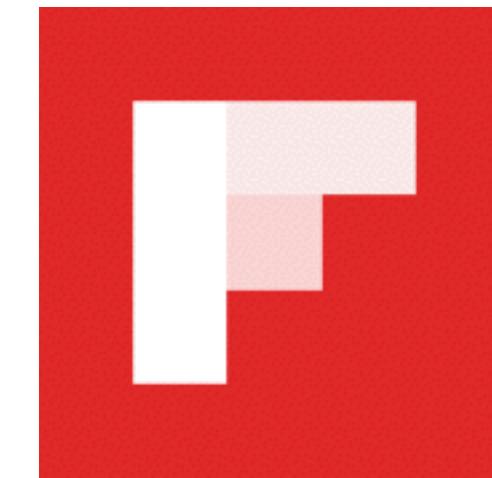
React Is A Terrible Idea | Lobsters
React Is A Terrible Idea browsers mobile web pandastrike.com. ... React may be a awful, terrible, and bring about the end of the world (free web), ...
<https://lobste.rs/s/wgceja>

React Is A Terrible Idea - Echo JS
React Is A Terrible Idea at www.pandastrike.com ... Both React and Web Components are overrated in my opinion. React is 127KB, Web Components polyfill is 105KB.
echojs.com/news/13855



NETFLIX

zapier*



BBC

Atlassian

● React
JavaScript library

● jQuery
Software

+ Add comparison

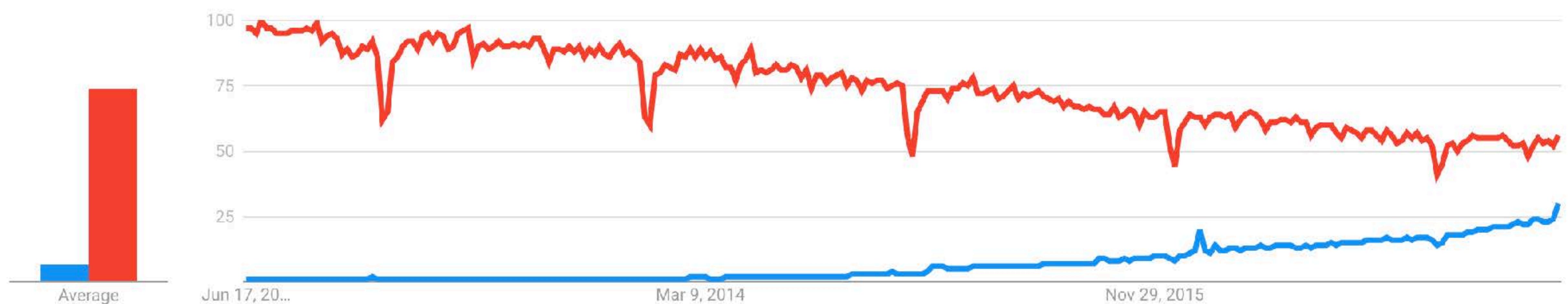
Worldwide ▾

Past 5 years ▾

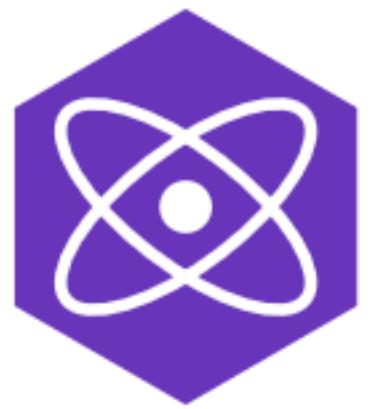
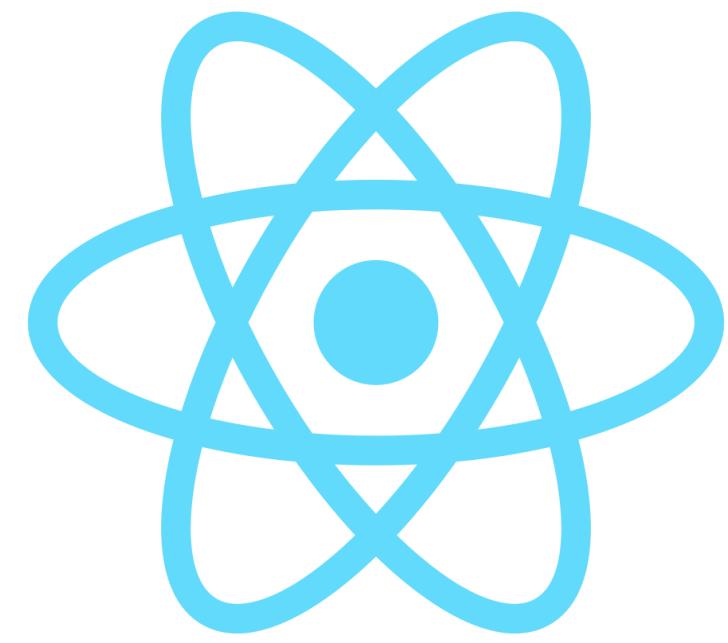
All categories ▾

Web Search ▾

Interest over time ?

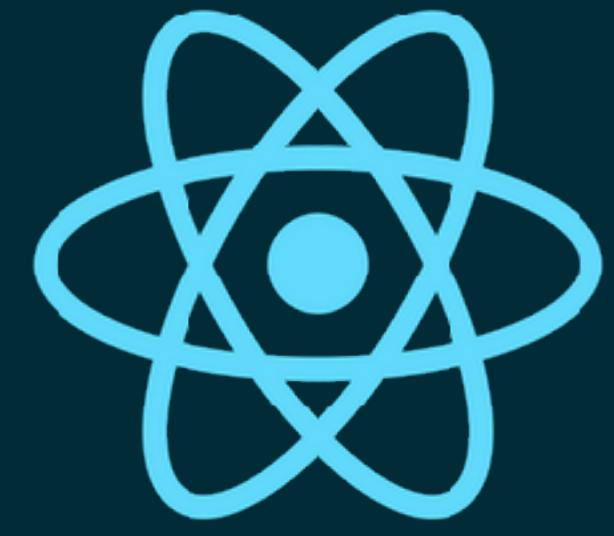


<https://trends.google.com/trends/explore?q=%2Fm%2F012l1vxv,%2Fm%2F0268gyp>



PREACT

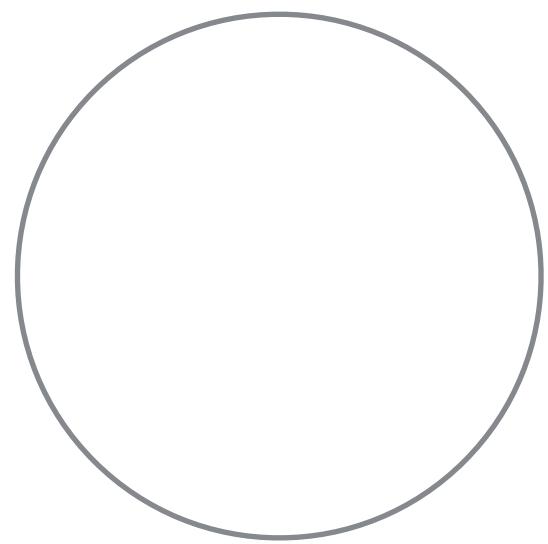




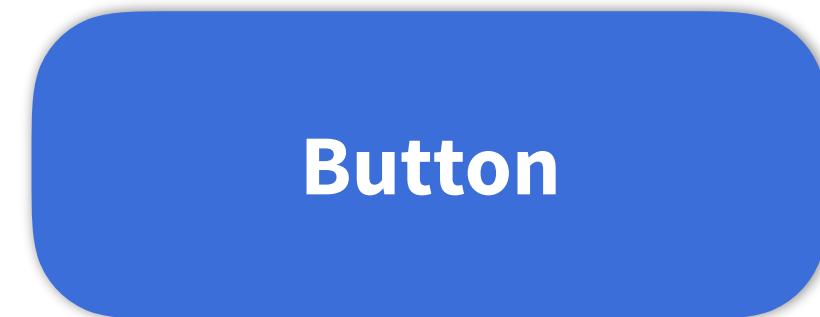
Everything is a component!

f(state) ⇒ UI

What about styles?



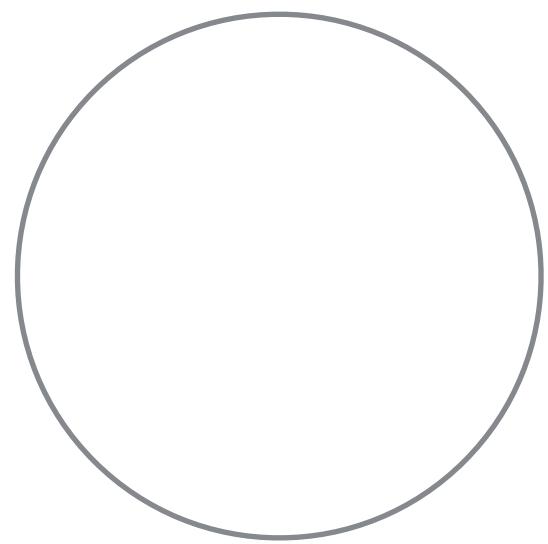
- Logic
- HTML
- Styles



- Logic
- HTML
- Styles



- Logic
- HTML
- Styles



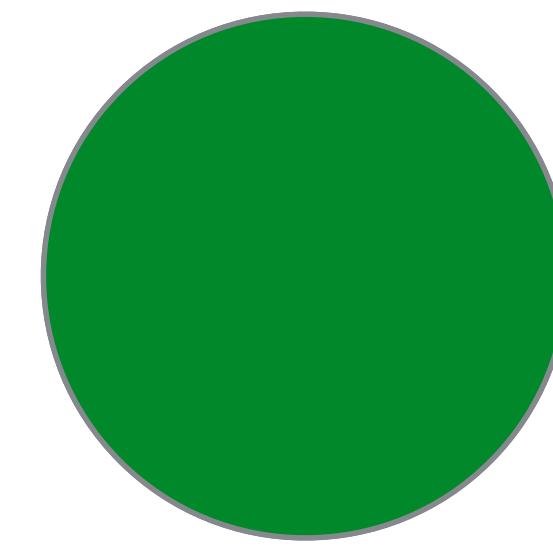
- Logic
- HTML
- Styles



- Logic
- HTML
- Styles



- Logic
- HTML
- Styles



- Logic
- HTML



- Logic
- HTML



- Logic
- HTML

Global styles



Global CSS

```
1 .btn {  
2   /* styles for button */  
3 }  
4  
5 .active {  
6   /* styles for active button */  
7   background-color: blue;  
8 }  
9  
10 .label {  
11   /* styles for button label */  
12 }
```

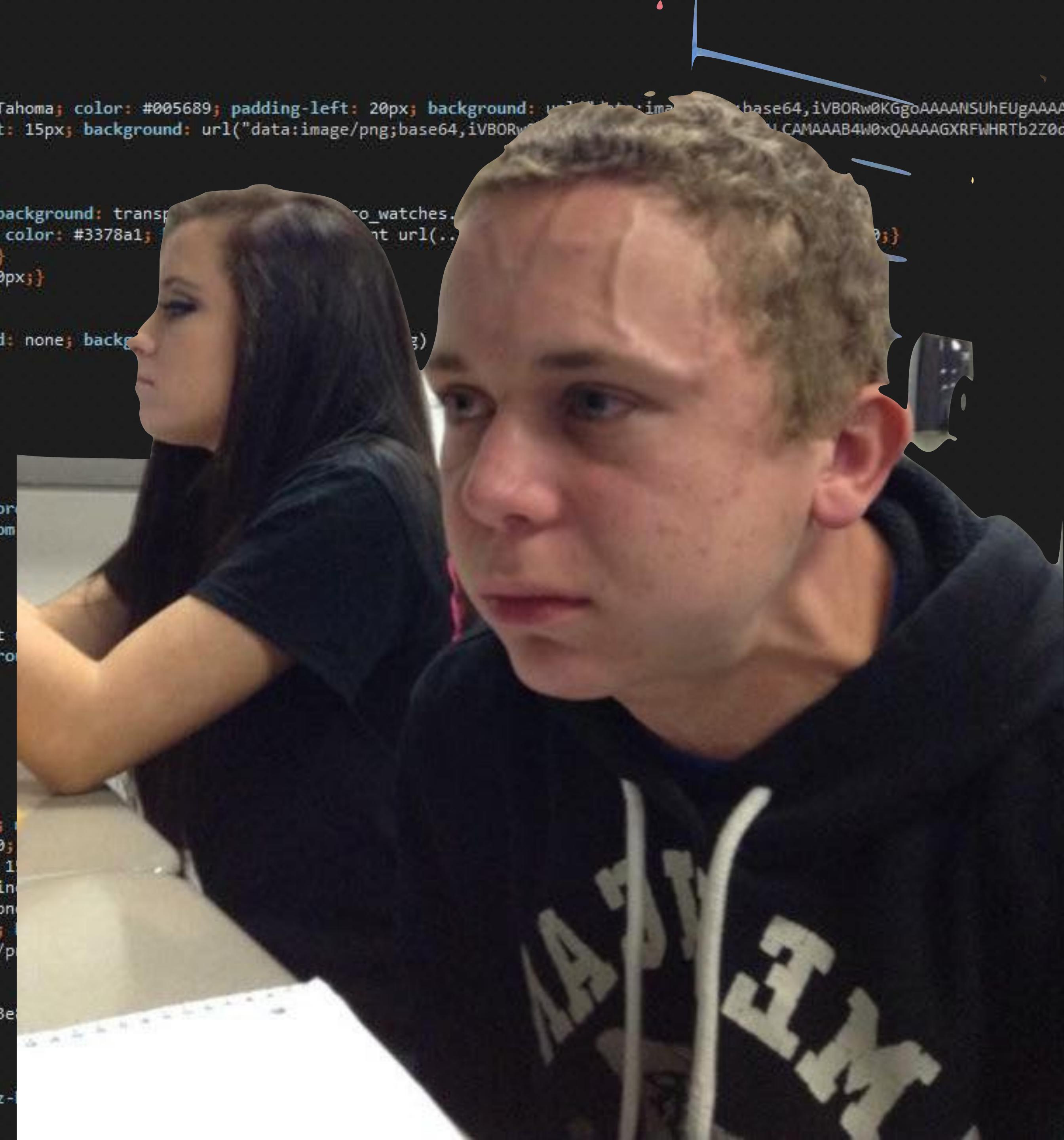


```
1 .star {  
2   /* styles for star */  
3 }  
4  
5 .active {  
6   /* styles for active star */  
7   background-color: orange;  
8 }  
9
```

.btn.active

.content .albums .album .btn

```
619 /* ----- Гостевая логика ----- */
620 div#guest-logic.block {}
621 #content div#guest-logic {float: left; width: 100%;}
622
623 div#guest-logic blockquote.subblock {padding: 0 18px; margin: 20px 0;}
624 div#guest-logic blockquote.subblock h3.entry-title {margin: 0 0 0.3em; font: bold 1.09em Tahoma; color: #005689; padding-left: 20px; background: url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
625 div#guest-logic blockquote.subblock div.entry-summary {margin-bottom: 0.5em; padding-right: 15px; background: url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
626 div#guest-logic blockquote.subblock div.entry-summary p {margin: 1em 0 0}
627 div#guest-logic blockquote.subblock div.entry-summary p:first-child {margin: 0}
628 div#guest-logic blockquote.subblock var.updated {position: absolute; visibility: hidden;}
629 div#guest-logic blockquote.subblock ul.details li a.details-watches {padding-left: 15px; background: transparent url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
630 div#guest-logic blockquote.subblock ul.details li a.details-comments {padding-left: 20px; color: #3378a1; background: transparent url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
631 div#guest-logic blockquote.subblock ul.details li a.details-comments var {color: #3378a1;}
632 #content div#guest-logic ul.pager {border-top: none; margin-bottom: 10px; padding-bottom: 20px;}
633
634 div#guest-logic ul#list-of-issues li a {color: #000; text-decoration: none;}
635 div#guest-logic ul#list-of-issues li var a {padding-right: 5px; color: #74060b; background: none; background-color: transparent; border: 1px solid #74060b; border-radius: 5px; width: 15px; height: 15px; display: inline-block; vertical-align: middle; margin-right: 10px; text-decoration: none; font-size: 0.8em; font-weight: bold; text-align: center; line-height: 1.2; font-family: sans-serif; transition: border 0.2s ease-in-out; z-index: 1; position: relative; }
636
637 /* ----- Репортер ----- */
638 div#reporting.block {}
639 #content div#reporting {float: right; width: 48%;}
640
641 div#reporting div.subblock {margin: 0 0 25px;}
642 div#reporting div.subblock img.photo {float: left; width: 100px; margin: 0 20px 10px 0; border: 1px solid #74060b; border-radius: 50%; background: transparent; }
643 div#reporting div.subblock h3.entry-title {margin-top: 0; margin-left: 120px; margin-bottom: 10px; font-size: 1.1em; font-weight: bold; color: #005689; }
644 div#reporting div.subblock div.entry-summary {margin-left: 120px; }
645 div#reporting div.subblock div.entry-summary p {margin: 0 0 0.2em; }
646 div#reporting div.subblock var.updated {position: absolute; visibility: hidden; }
647 div#reporting ul.details {margin: 0 0 0 120px; }
648 div#reporting ul.details li a.details-watches {padding-left: 15px; background: transparent url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
649 div#reporting ul.details li a.details-comments {padding-left: 20px; color: #3378a1; background: transparent url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
650 div#reporting ul.details li a.details-comments var {color: #3378a1; }
651
652 /* ----- Самое читаемое ----- */
653 div#most-readable.block {}
654 #content div#most-readable {clear: both; float: left; width: 46%;}
655
656 div#most-readable h2 {margin-bottom: 20px; }
657 div#most-readable ul#tab_most-readable {position: relative; bottom: -1px; overflow: hidden; border-bottom: 1px solid #74060b; }
658 div#most-readable ul#tab_most-readable li {float: left; height: auto; margin: 0; padding: 0; }
659 div#most-readable ul#tab_most-readable li a {position: relative; display: block; z-index: 1; }
660 div#most-readable ul#tab_most-readable li a:hover {color: #74060b; text-decoration: underline; }
661 div#most-readable ul#tab_most-readable li.tabs-selected a {z-index: 20; text-decoration: none; }
662 div#most-readable ul#tab_most-readable li a dfn {display: block; padding: 4px 20px 7px 4px; border: 1px solid #74060b; border-radius: 5px; }
663 div#most-readable ul#tab_most-readable li.tabs-selected a dfn {background: url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA
664
665
666 div#most-readable div.subblock {margin: 0 20px 0 0; padding: 20px; border: 2px solid #d9e3e9; border-radius: 10px; }
667 div#most-readable div.tabs-container { visibility: visible; z-index: 5; }
668 div#most-readable div.tabs-hide {visibility: hidden; position: absolute; }
669
670 div#most-readable div.subblock, x:-moz-any-link, x:default { -moz-border-radius: 4px; -moz-
671
672
673 div#most-readable div.subblock h3 {position: absolute; visibility: hidden; }
```





BEM

<https://en.bem.info/>

Block

.Block

Element

.Block--element

Modifier

.Block--element__modifier

Without BEM

```
<button class="button active">
  <span class="label">
    click me!
  </span>
</button>
```

With BEM

```
<button class="Button Button__active">
  <span class="Button--label">
    click me!
  </span>
</button>
```

FOOD GROUPS

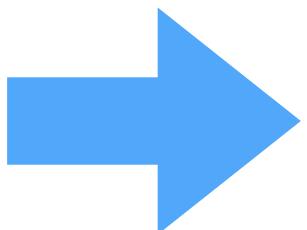


BEM prevents this!

Manual work

```
<ul class="nav">
  <li class="nav__item nav__item_active"><a class="nav__link">One</a></li>
  <li class="nav__item"><a class="nav__link">Two</a></li>
  <li class="nav__item"><a class="nav__link">Three</a></li>
</ul>
```

```
.nav__item {  
  padding: 4px 10px;  
  color: black;  
}  
  
.nav__item_active {  
  font-weight: bold;  
  background: #ffc7c7;  
}
```



```
.navigation__item {  
  padding: 4px 10px;  
  color: black;  
}  
  
.navigation__item_active {  
  font-weight: bold;  
  background: #ffc7c7;  
}
```

@@ -2,41 +2,41 @@

```
2     <fest:script>
3         console.log('json', json);
4     </fest:script>
5     - <div class="content_game-menu content_game-menu_register">
6         <div class="game-menu">
7             <ul class="game-menu_nav">
8                 <form class="game-menu_form" action=""
9                     novalidate="novalidate" autocomplete="off">
10                    <li class="game-menu_nav-element">
11                        <input class="game-menu_nav-item game-
12                         menu_item_input js-focus" name="login" type="text" required="required" maxlength="20"
13                         placeholder="LOGIN"/>
14                    </li>
15                    <li class="game-menu_nav-element">
16                        <input class="game-menu_nav-item game-
17                         menu_item_input js-focus" name="email" type="email" required="required" maxlength="20"
18                         placeholder="E-MAIL"/>
19                    </li>
20                    <li class="game-menu_nav-element">
21                        <input class="game-menu_nav-item game-
22                         menu_item_input js-focus" name="password" type="password" required="required"
23                         maxlength="10" placeholder="PASSWORD"/>
24                    </li>
25                    <li class="game-menu_nav-element">
26                        <div class="game-menu_nav-avatar">
27                            <div class="nav-avatar">
28                                <div class="nav-
29                                   avatar_icons">
30                                    <i class="fa fa-
31                                       trash-o nav-avatar_delete nav-avatar_delete_invisible js-delete-avatar" title="Delete
32                                       this pic!"></i>
33                                    <i class="fa fa-
34                                       camera nav-avatar_camera js-camera" title="Take a pic now!"></i>
```

```
2     <fest:script>
3         console.log('json', json);
4     </fest:script>
5     + <div class="game_menu-wrapper game_menu-wrapper_register">
6         + <div class="game_menu">
7             + <ul class="menu">
8                 + <form class="menu_form js-form" action=""
9                     novalidate="novalidate" autocomplete="off">
10                    <li class="menu_element">
11                        <input class="menu_item menu_item_input
12                         js-focus" name="login" type="text" required="required" maxlength="20" placeholder="LOGIN"/>
13                    </li>
14                    <li class="menu_element">
15                        <input class="menu_item menu_item_input
16                         js-focus" name="email" type="email" required="required" maxlength="20" placeholder="E-
17                         MAIL"/>
18                    </li>
19                    <li class="menu_element">
20                        <input class="menu_item menu_item_input
21                         js-focus" name="password" type="password" required="required" maxlength="10"
22                         placeholder="PASSWORD"/>
23                    </li>
24                    <li class="menu_element">
25                        <div class="menu_avatar">
26                            <div class="avatar">
27                                <div class="avatar_icons">
28                                    <i class="fa fa-
29                                       trash-o avatar_delete avatar_delete_invisible js-delete-avatar" title="Delete this pic!"></i>
30                                    <i class="fa fa-
31                                       camera avatar_camera js-camera" title="Take a pic now!"></i>
```



WORK HARD

Manual work

css MODULES

<https://github.com/css-modules/css-modules>

Before: BEM-style

```
const Button = ({ children }) => (
  <button className="Button">
    <span className="Button__label">
      { children }
    </span>
  </button>
)
```

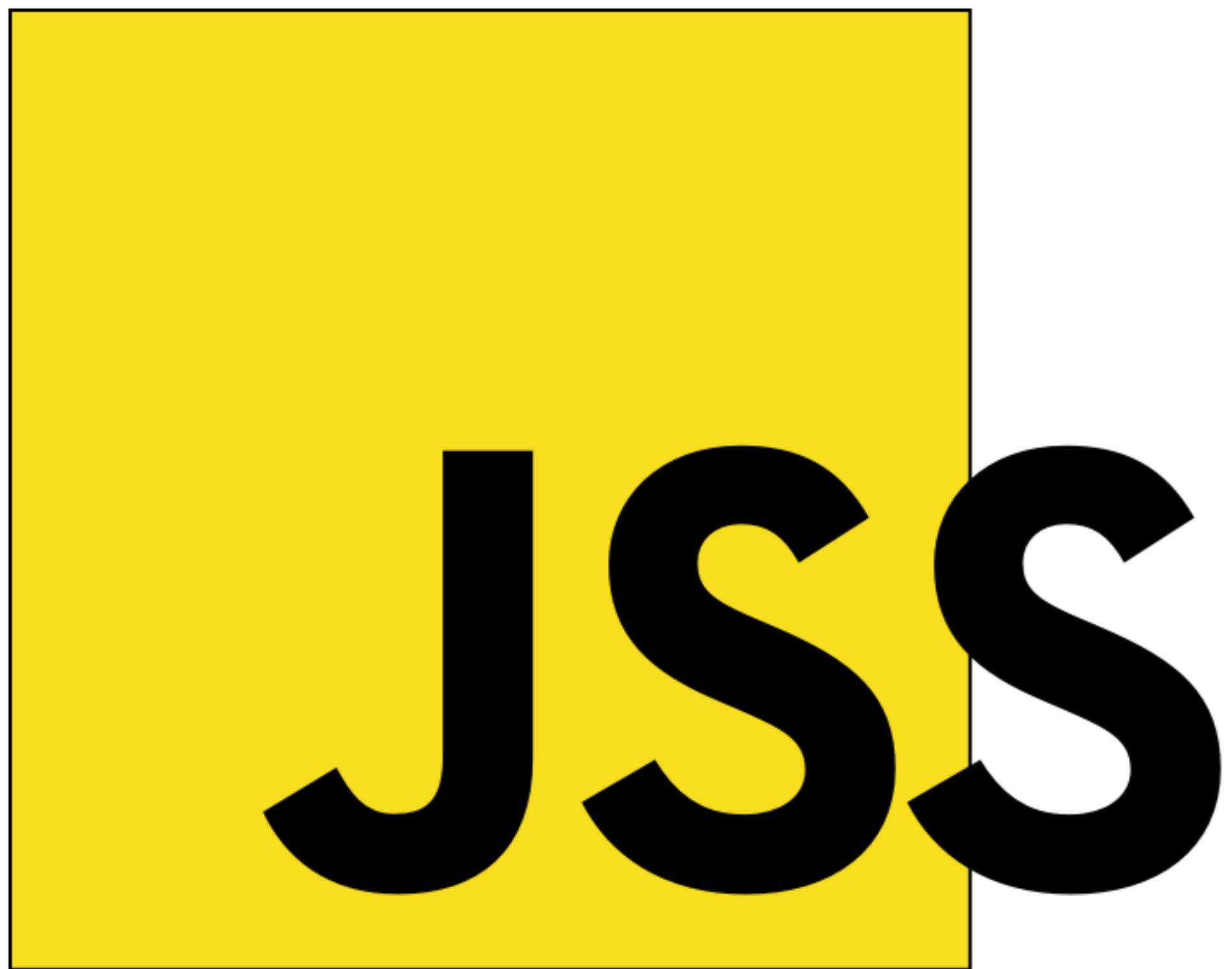
After: CSS-modules

```
import styles from './Button.css'

const Button = ({ children }) => (
  <button className={styles.button}>
    <span className={styles.label}>
      { children }
    </span>
  </button>
)
```

CSS-modules

- ✓ Explicit imports
- ✓ Scoped & fast selectors
- ✓ True rules isolation
- ✓ Code reuse, expressiveness
- ✓ Framework agnostic
- ✗ Non standard syntax (compose, vals, etc.)
- ✗ Build step is required
- ✗ No dead code elimination
- ✗ No automatic vendor prefixing



<https://cssinjs.org>

```
export const styles = {
  button: {
    padding: '10px',
    '&:hover': {
      background: 'blue'
    }
  },
  '@media (min-width: 1024px)': {
    button: {
      padding: '20px'
    }
  }
}
```

Before: BEM-style

```
const Button = ({ children }) => (
  <button className="Button">
    <span className="Button__label">
      { children }
    </span>
  </button>
)
```

After: JSS

```
import injectSheet from 'react-jss'
import styles from './styles'

const Button = ({ classes, children }) => (
  <button className={classes.button}>
    <span className={classes.label}>
      {children}
    </span>
  </button>
)

export default injectSheet(styles)(Button)
```

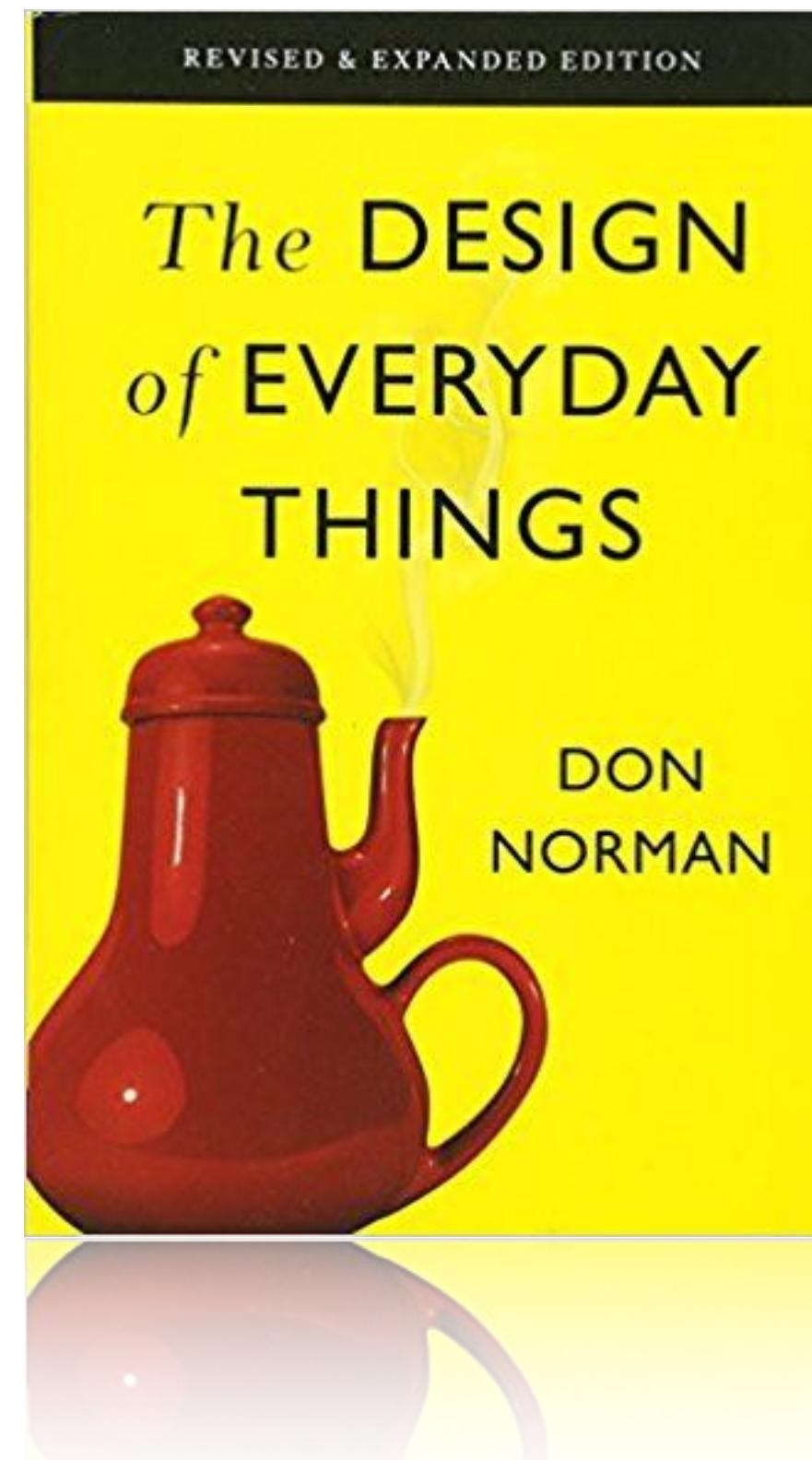
CSS-in-JS (JSS)

- ✓ Explicit imports
- ✓ Scoped & fast selectors
- ✓ True rules isolation
- ✓ Code reuse, expressiveness
- ✓ Framework agnostic
- ✓ Uses w3c standard
- ✓ No build step is required
- ✓ Dead code elimination
- ✓ Automatic vendor prefixing

“If we generate class names,
why do we still use class attribute?”

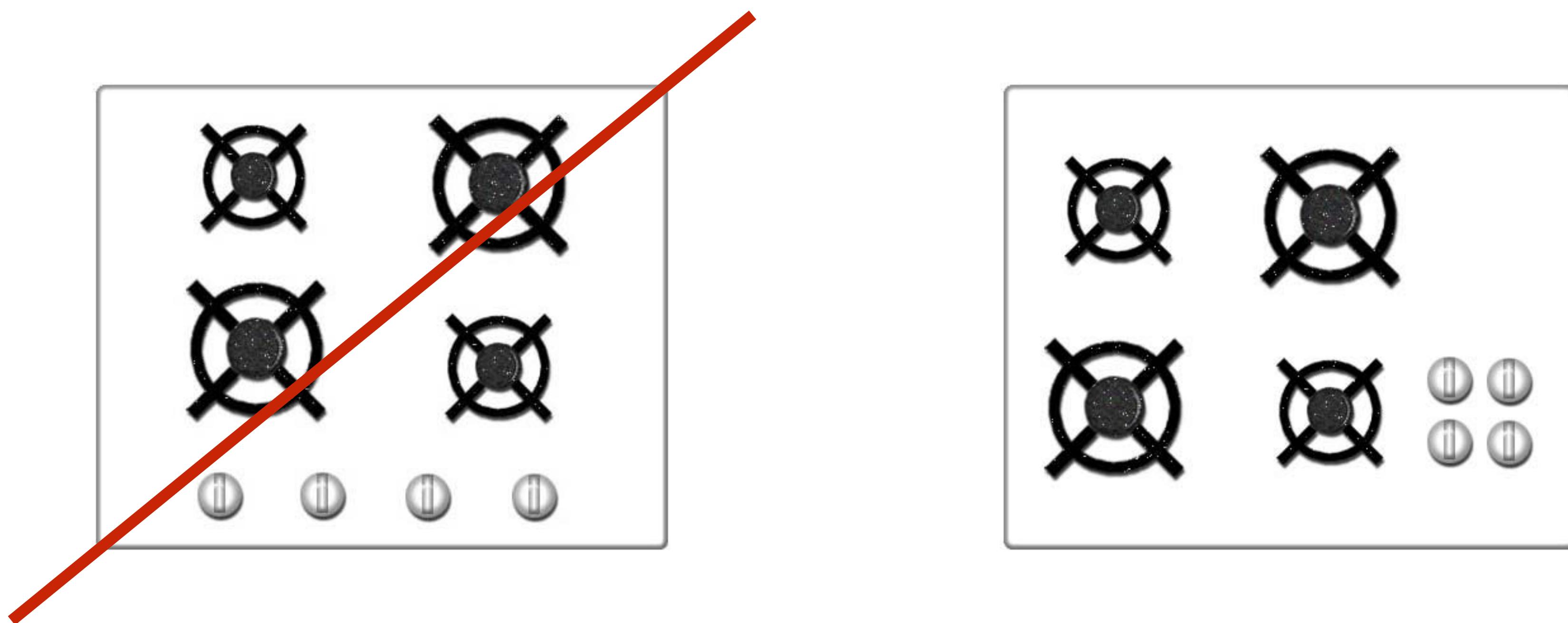
–Max or Glenn, probably?

Natural mapping



The Design of Everyday Things
by Don Norman

Natural mapping



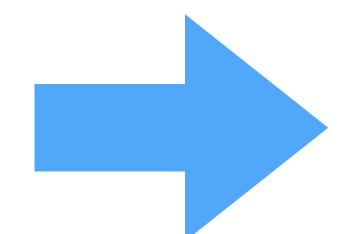


**styled
components**

<https://www.styled-components.com/>

```
// Create a Title component that'll render
// an <h1> tag with some styles
const Title = styled.h1`  
  font-size: 1.5em;  
  text-align: center;  
  color: palevioletred;  
`;  
  
render(  
  <Title>  
    Hello PiterCSS!  
  </Title>  
);
```

```
const Button = styled.button`  
  /* Adapt the colours based on primary prop */  
  background: ${props => props.primary ? 'palevioletred' : 'white'};  
  color: ${props => props.primary ? 'white' : 'palevioletred'};  
  font-size: 1em;  
  margin: 1em;  
  padding: 0.25em 1em;  
  border: 2px solid palevioletred;  
  border-radius: 3px;  
`;  
  
render(  
  <div>  
    <Button>Normal</Button>  
    <Button primary>Primary</Button>  
  </div>  
)
```



“styled-components [...] removes the mapping between components and styles.”

<https://www.styled-components.com/>



styled-components

- ✓ No messing with classNames (implementation detail)
- ✓ Same mental model and structure for the whole application

**And because the model is
the same...**



styled-components for react-native!

```
import styled from 'styled-components/native';

const StyledView = styled.View`  
  background-color: papayawhip;  
`;  
  
const StyledText = styled.Text`  
  color: palevioletred;  
`;  
  
class MyReactNativeComponent extends React.Component {  
  render() {  
    return (  
      <StyledView>  
        <StyledText>Hello World!</StyledText>  
      </StyledView>  
    )  
  }  
}
```

“No UI is about machines helping us, instead of us adapting for computers.”

– <https://www.cooper.com/journal/2012/08/the-best-interface-is-no-interface>

Myths & Lies

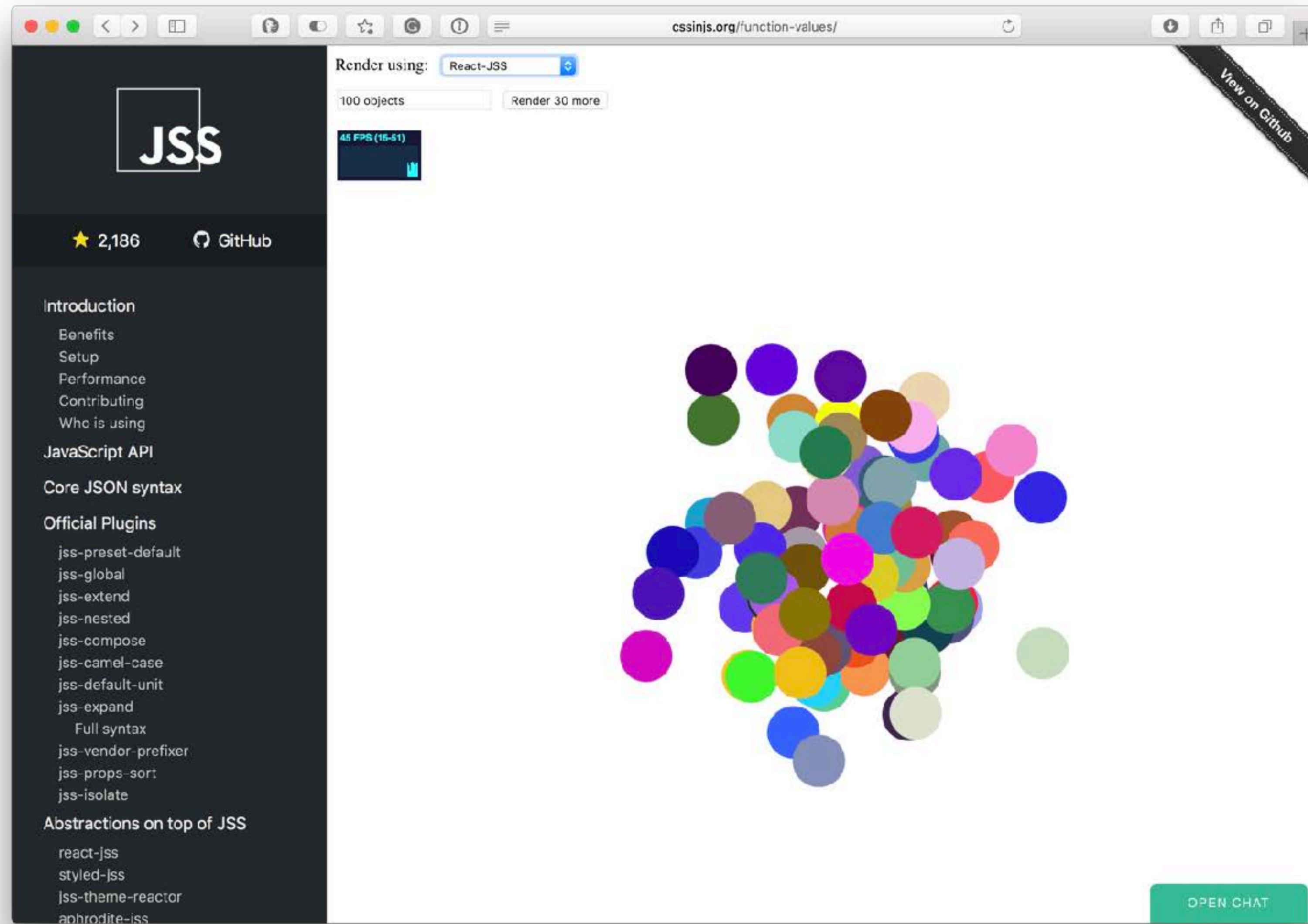
CSS-in-JS is slow

CSS-in-JS ≠ inline styles!

```
import { css } from 'glamor'

const title = css({
  fontSize: '1.8em',
  fontFamily: 'Comic Sans MS',
  color: 'blue'
})

console.log(title)
// → 'css-1pyvz'
```



<http://cssinjs.org/function-values/>

Maintanability > Speed

▲ bigmanwalter 11 hours ago [-]

▼ The React documentation says it so it has to be true. DOM slow, VDOM fast. i
Facebook is out there buying developer mindshare.

[reply](#)

* 1 point by danabramov [1 minute ago](#) | [edit](#) | [delete](#) [-]

We don't claim that in React documentation. React can't be faster than the same DOM mutations written by hand because by definition it has to do more work.

We do think it helps to create maintainable apps though, and it is *fast enough* for practical use cases despite its immutable design. That's why we use it at Facebook.

[reply](#)



Dan Abramov

@dan_abramov

[Follow](#)

Myth: React is “faster than DOM”. Reality: it helps create maintainable applications, and is **fast enough** for most use cases.

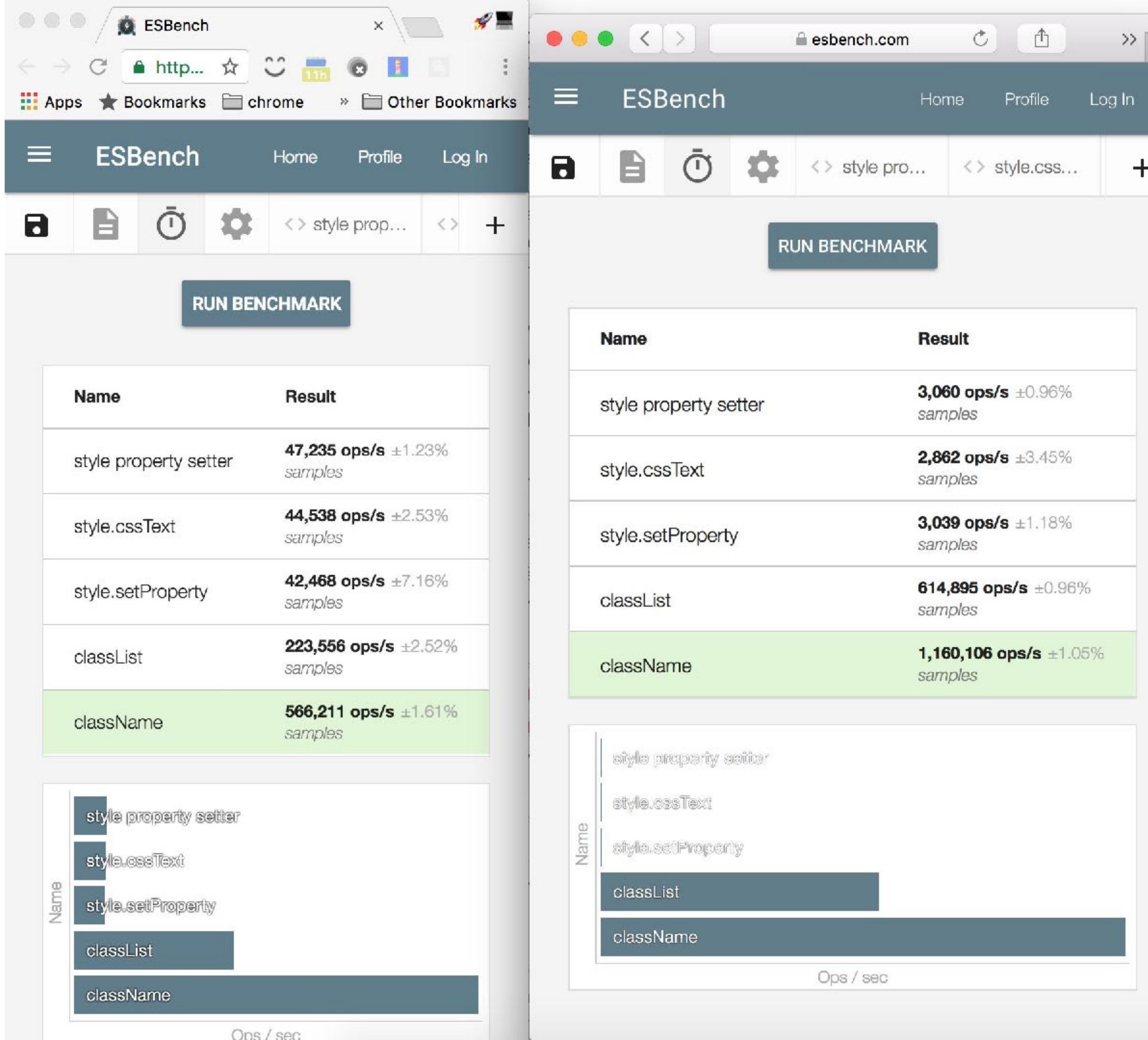
12:01 PM - 16 Mar 2017



366

791

https://twitter.com/dan_abramov/status/842329893044146176





James Kyle 

@thejameskyle

Following



A point a lot of people miss about CSS-in-JS is that the perf problem is fixable, with stylesheets the maintainability/scaling issues are not

RETWEETS

49

LIKES

159



2:52 PM - 4 May 2017



The screenshot shows a browser window with the URL codepen.io/malyw/pen/XRRgeB?editors=0110. The page title is "CSS site theme from CSS Custom Properties" by Serg Hospodarets PRO. The interface is split into two main sections: "CSS" on the left and "JS" on the right.

HTML section (CSS):

```
:root {  
  --base-color: #fff;  
  --main-r: 255;  
  --main-g: 255;  
  --main-b: 255;  
  --is-in-light-half-r: 1;  
  --is-in-light-half-g: 1;  
  --is-in-light-half-b: 1;  
  --main-base-color: rgb(var(--main-r), var(--main-g), var(--main-b));  
  --main-bg-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 16)), calc(var(--main-g) - (var(--is-in-light-half-g) * 172)), calc(var(--main-b) - (var(--is-in-light-half-b) * 175));  
  --main-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 204)), calc(var(--main-g) - (var(--is-in-light-half-g) * 204)), calc(var(--main-b) - (var(--is-in-light-half-b) * 204));  
  --main-bg: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 23)), calc(var(--main-g) - (var(--is-in-light-half-g) * 21)), calc(var(--main-b) - (var(--is-in-light-half-b) * 9));  
  --header-bg: var(--main-bg-color);  
  --navlinks-border-color: rgba(var(--main-r), var(--main-g), var(--main-b), .2);  
  --author-name-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 200)), calc(var(--main-g) - (var(--is-in-light-half-g) * 184)), calc(var(--main-b) - (var(--is-in-light-half-b) * 176));  
  --avatar-shadow-color: rgba(calc(var(--main-r) - (var(--is-in-light-half-r) * 255)), calc(var(--main-g) - (var(--is-in-light-half-g) * 255)), calc(var(--main-b) - (var(--is-in-light-half-b) * 255)), .6);  
  --nav-links-color: var(--main-base-color);  
  --nav-links-bg: var(--main-bg-color);  
  --post-bg: var(--main-base-color);  
  --pub-date-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 85)), calc(var(--main-g) - (var(--is-in-light-half-g) * 85)), calc(var(--main-b) - (var(--is-in-light-half-b) * 85));  
  --post-shadow-color: rgba(calc(var(--main-r) - (var(--is-in-light-half-r) * 255)), calc(var(--main-g) - (var(--is-in-light-half-g) * 255)), calc(var(--main-b) - (var(--is-in-light-half-b) * 255)), .4);  
}  
body, h1, h2, h3, h4, h5, h6, html, img, p, ul {  
  margin: 0;  
}
```

JS section (JavaScript):

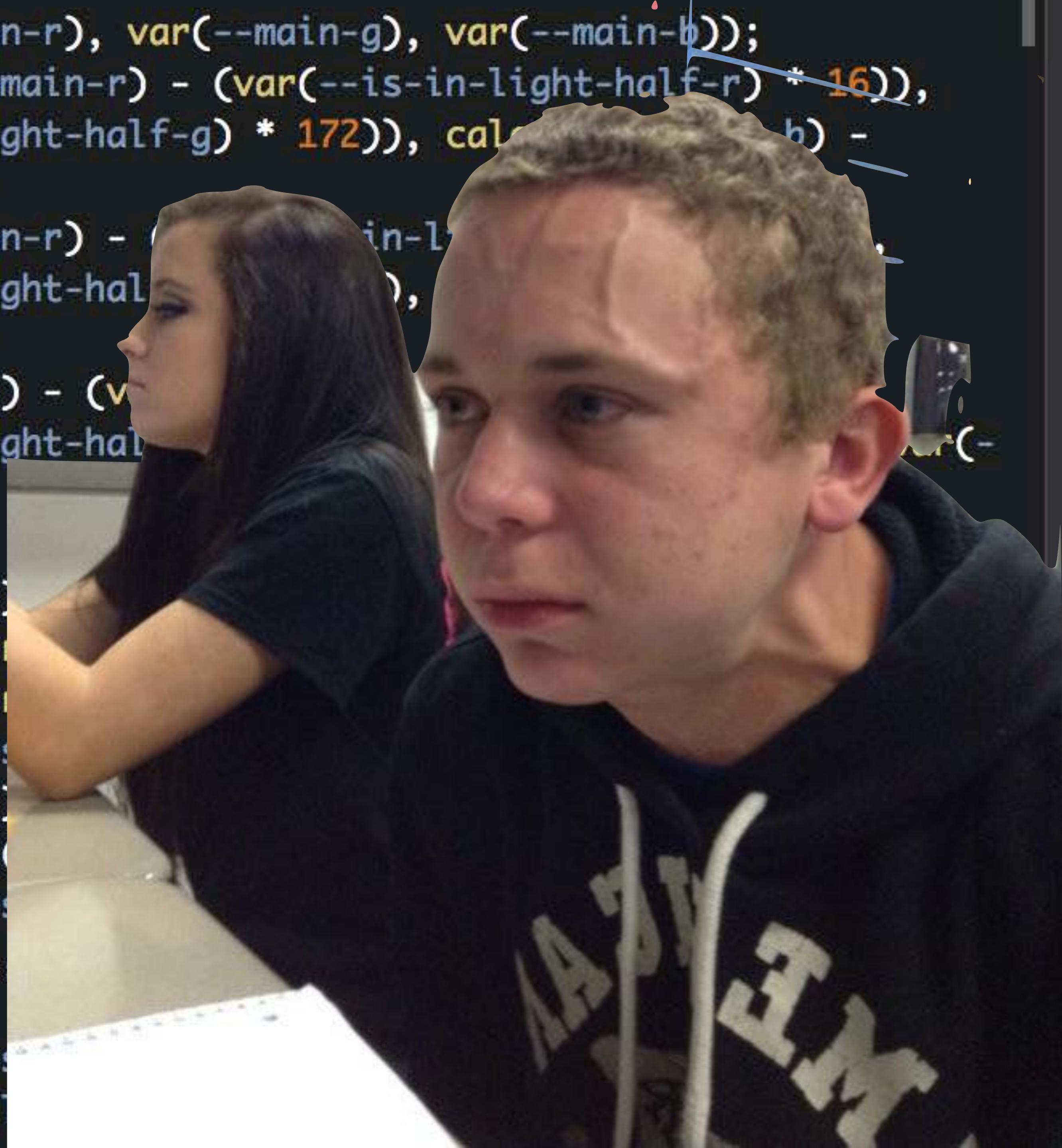
```
/* => {  
  /*  
   * Listen base color change and assign global CSS variables from it  
   */  
  
  function assignGlobalProp(name, value) {  
    document.documentElement.style.setProperty(`--main-${name}`, value);  
    document.documentElement.style.setProperty(`--is-in-light-${name}`, value > 128 ? 1 : -1);  
  }  
  
  const baseColorInput = document.querySelector('.base-color');  
  
  baseColorInput.addEventListener("input", function O {  
    changeMainColors(baseColorInput.value);  
  });  
  
  function changeMainColors(baseColorHex) {  
    // expected color is rgb(r,g,b) or rgba(r,g,b,a)  
    const r = parseInt(baseColorHex.substring(1, 3), 16);  
    const g = parseInt(baseColorHex.substring(3, 5), 16);  
    const b = parseInt(baseColorHex.substring(5, 7), 16);  
  
    assignGlobalProp('r', r);  
    assignGlobalProp('g', g);  
    assignGlobalProp('b', b);  
  }  
  
  // INIT  
  //changeMainColors(baseColorInput.value);  
  
  // OPTIONS  
  const options = {};  
  
  let optionEls = document.querySelectorAll('.options input');  
  optionEls = [...optionEls];  
  
  optionEls.forEach((optionEl) => {  
    // use DOM classes as option names  
    const optionName = optionEl.className;  
  
    // TNTT  
  });  
}
```

At the bottom of the editor, there is a color picker labeled "Pick base color" with the hex value `ffffff`. Below the color picker are buttons for "Console", "Assets", "Comments", and "Keyboard". To the right are buttons for "Share", "Export", and "Embed".

<http://slides.com/malyw/houdini-codemotion#/16>

```
9  --main-base-color: rgb(var(--main-r), var(--main-g), var(--main-b));  
10 --main-bg-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 16)),  
11 calc(var(--main-g) - (var(--is-in-light-half-g) * 172)), calc(var(--main-b) -  
12 (var(--is-in-light-half-b) * 175)));  
13 --main-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 16)),  
14 calc(var(--main-g) - (var(--is-in-light-half-g) * 172)), calc(var(--main-b) -  
15 (var(--is-in-light-half-b) * 204)));  
16 --main-bg: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 16)),  
17 calc(var(--main-g) - (var(--is-in-light-half-g) * 172)), calc(var(--main-b) -  
18 (var(--is-in-light-half-b) * 9)));
```

```
19 --header-bg: var(--main-bg-color);  
20 --navlinks-border-color: rgba(var(--main-base-color), 1, 0, 0);  
21 --author-name-color: rgb(calc(var(--main-r) - (var(--is-in-light-half-r) * 200)), calc(var(--main-g) - (var(--is-in-light-half-g) * 176)), calc(var(--main-b) - (var(--is-in-light-half-b) * 176)));  
22 --avatar-shadow-color: rgba(calc(var(--main-base-color), 1, 0, 0), 255, 255, 255);  
23 --avatar-shadow-color: rgba(calc(var(--main-base-color), 1, 0, 0), 255, 255, 255), calc(var(--main-g) - (var(--is-in-light-half-g) * 172)), calc(var(--main-b) - (var(--is-in-light-half-b) * 255)));  
24 --nav-links-color: var(--main-base-color);  
25 --nav-links-bg: var(--main-bg-color);
```



```
import React from 'react'
import { createComponent } from 'react-fela'
import felaSnapshot from './test-helpers/felaSnapshot.js'

const boxRules = ({ size = 10 }) => ({
  width: size + 'px',
  height: size + 'px',
  backgroundColor: 'red'
})

const Box = createComponent(boxRules)

describe('Box', () => {
  it('should render component', () => {
    const snapshot = felaSnapshot(<Box>hello</Box>)
    expect(snapshot).toMatchSnapshot()
  })
})
```

```
exports[`Box should change box size when size prop is passed 1`] = `>
<div
  className="a b c"
  id={undefined}
  style={undefined}>
  hello
</div>
`;
```

**Crafted CSS is better for
performance**

meiert.com/en/blog/20170531/70-percent-css-repetition/

Home **Archive** **Biography** **Contact** **Search** **Deutsch**

70% Repetition in Style Sheets: Data on How We Fail at CSS Optimization

Post from May 31, 2017, reflecting Jens the [Developer](#).

Teaser: Check on how many declarations you use in your style sheets, how many of those declarations are unique, and what that means.

In 2008 I've argued that [using declarations just once](#) makes for a key method to DRY up our style sheets (this works because avoiding declaration repetition is *usually* more effective than avoiding selector repetition—declarations are often longer). I've later raised the suspicion that the demand for variables would have been more informed and civilized had we [nailed style sheet optimization](#). What I haven't done so far is gather data to *show* how big the problem really is. That's what I've done in the last few weeks.

In a [Google spreadsheet](#) I've collected the Top 200 of content sites in the [The Moz Top 500](#), and taken another 20 sites related to web development for comparison. (I've also added 3 of my sites out of curiosity.) I've then used the extremely useful [CSS Stats](#) to determine the total number of CSS declarations, as well as the number of *unique* declarations, to calculate ratios as well as averages: You get the idea as soon as you check out said [spreadsheet](#).

Website	Average	Total Declarations	Unique Declarations	Ratio Unique Total
1. https://www.decodoku.com/	1133	1811	939	0.52
2. https://cheatsheet.com/	1720	3063	1063	0.35
3. https://www.gutenberg.org/	78	83	83	1.00
4. https://www.gutenberg.org/	4338	2717	2717	0.99
5. https://www.gutenberg.org/	2071	549	549	0.97
6. https://www.gutenberg.org/	1635	996	996	0.99
7. https://www.gutenberg.org/	1867	996	996	0.99
8. https://www.gutenberg.org/	2952	1212	1212	0.99
9. https://www.gutenberg.org/	5000	1983	1983	0.99
10. https://www.gutenberg.org/	5271	882	882	0.99
11. https://www.gutenberg.org/	3946	2398	2398	0.99
12. https://www.gutenberg.org/	5462	1729	1729	0.99
13. https://www.gutenberg.org/	1248	514	514	0.99
14. https://www.gutenberg.org/	3368	1592	1592	0.99
15. https://www.gutenberg.org/	21120	4858	4858	0.99
16. https://www.gutenberg.org/	9421	2881	2881	0.99
17. https://www.gutenberg.org/	5849	1994	1994	0.99
18. https://www.gutenberg.org/	1418	960	960	0.99
19. https://www.gutenberg.org/	1108	593	593	0.99
20. https://www.gutenberg.org/	4705	2269	2269	0.99

Leave a Comment

Name:

Website:

Comment:

Respect the [comment guidelines](#). HTML allowed:
 <blockquote> <code>

[preview](#)

Customer Sup
Be Known For Better Customer Support for Free Today.

This website uses cookies and is required by EU law to tell you.

[Wonderful](#)

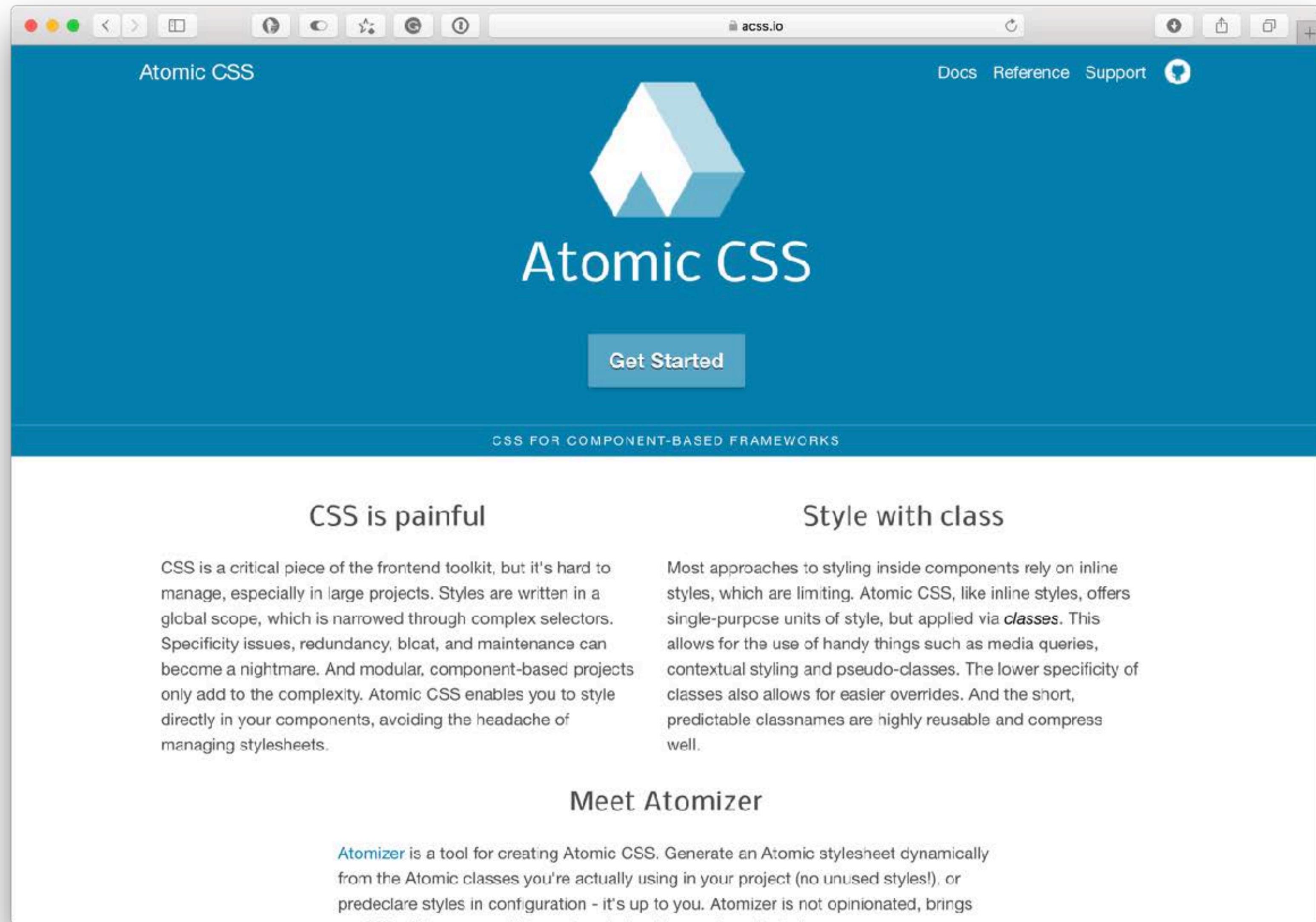
[Cookie Consent](#)

“In CSS, we repeat ourselves too much. While it’s absolutely, practically possible to limit declaration repetition to 10–20%, reality averages 72% (median 66%).”

<https://meiert.com/en/blog/20170531/70-percent-css-repetition/>



Use functional CSS



The screenshot shows a web browser window with the URL `acss.io` in the address bar. The page has a dark blue header with the text "Atomic CSS" on the left and "Docs Reference Support" on the right. A large, stylized white "A" logo is centered above the main title. The main title "Atomic CSS" is in a large, white, sans-serif font. Below the title is a blue button with the text "Get Started". Under the title, the text "CSS FOR COMPONENT-BASED FRAMEWORKS" is displayed in a smaller, white, sans-serif font. The main content area is white and contains two columns of text. The left column is titled "CSS is painful" and discusses the challenges of managing CSS in large projects. The right column is titled "Style with class" and explains how Atomic CSS uses classes for styling. At the bottom, there is a section titled "Meet Atomizer" with a description of its purpose. The overall design is clean and modern, using a combination of white, blue, and dark blue colors.

Atomic CSS

Docs Reference Support

CSS is painful

CSS is a critical piece of the frontend toolkit, but it's hard to manage, especially in large projects. Styles are written in a global scope, which is narrowed through complex selectors. Specificity issues, redundancy, bloat, and maintenance can become a nightmare. And modular, component-based projects only add to the complexity. Atomic CSS enables you to style directly in your components, avoiding the headache of managing stylesheets.

Style with class

Most approaches to styling inside components rely on inline styles, which are limiting. Atomic CSS, like inline styles, offers single-purpose units of style, but applied via *classes*. This allows for the use of handy things such as media queries, contextual styling and pseudo-classes. The lower specificity of classes also allows for easier overrides. And the short, predictable classnames are highly reusable and compress well.

Meet Atomizer

Atomizer is a tool for creating Atomic CSS. Generate an Atomic stylesheet dynamically from the Atomic classes you're actually using in your project (no unused styles!), or predeclare styles in configuration - it's up to you. Atomizer is not opinionated, brings

<https://acss.io>

Atomic CSS

```
<div class="Pos(a) Bgc(brandColor) W(columnWidth) H(90px)"></div>
<div class="C(brandColor) BdB Bdc(brandColor) Mstart(columnWidth) P(10px)">
  Lorem ipsum
</div>
```



Lea Verou

@LeaVerou

Follow



Just saw [@AtomicCSS](#)
Wow.

Syntax so bad I looked for "Is this a joke?" in the FAQ. Looks like a styling language designed by assembly coders!

RETWEETS

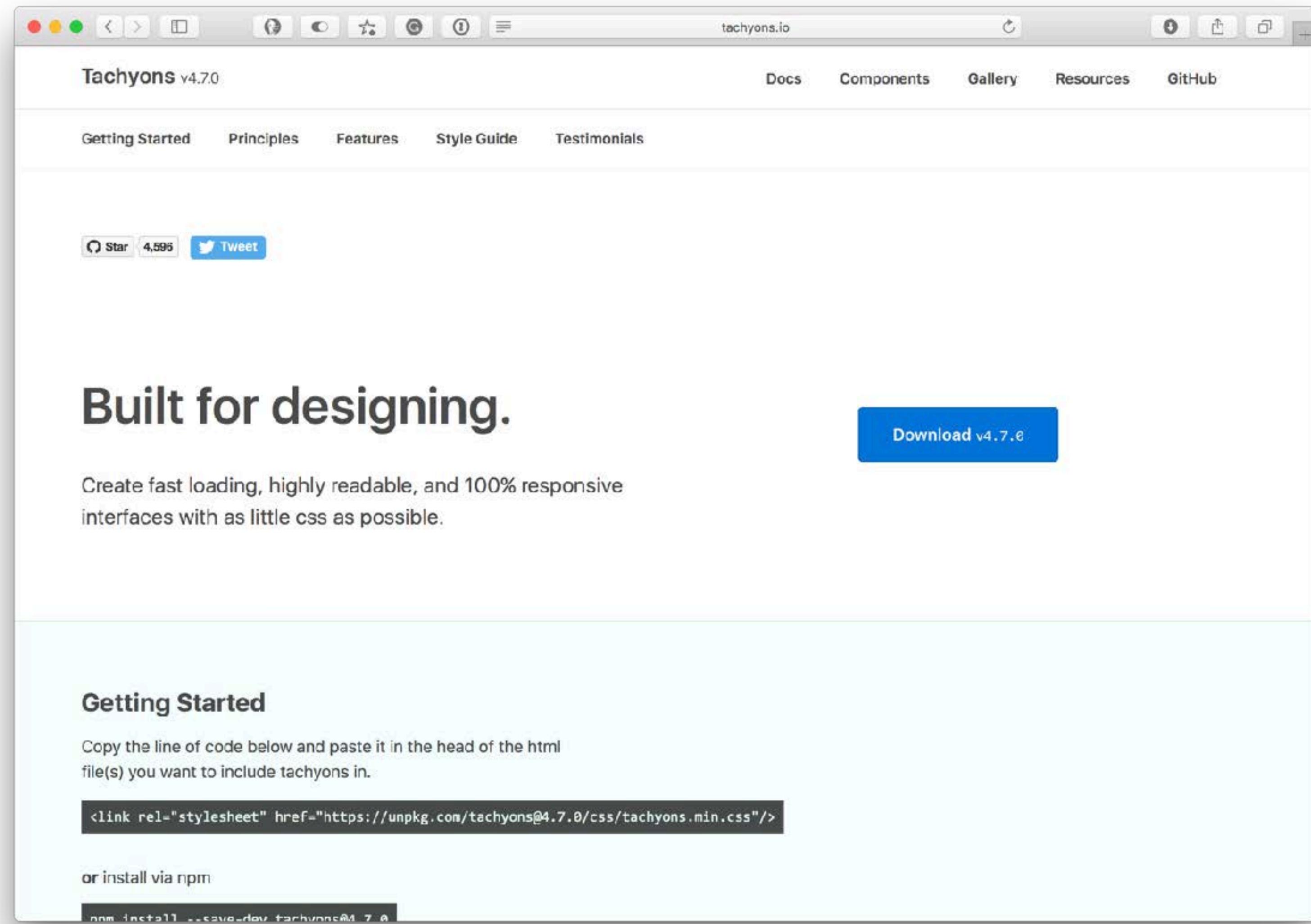
50

LIKES

87



9:50 PM - 11 Jan 2016



<http://tachyons.io>

Tachyons

```
<article class="pa3 pa5-ns">
  <h1 class="f3 f1-m f-headline-l">Title</h1>
  <p class="measure lh-copy">
    Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod
    tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At
    vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren,
    no sea takimata sanctus est Lorem ipsum dolor sit amet.
  </p>
</article>
```



Dave Shea

@mezzoblue

Follow

Huh, just came across github.com/tachyons-css/tachyons for the first time.

Worst web dev idea of all time, or just a really bad one?



tachyons-css/tachyons

Functional css for humans. Contribute to tachyons development by creating an account on GitHub.

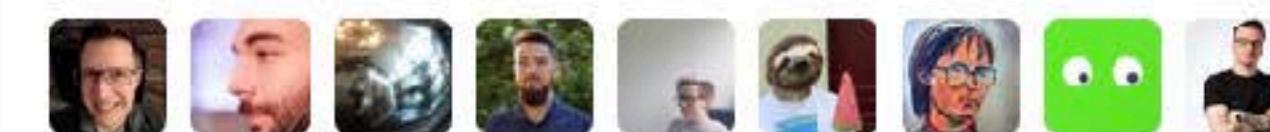
github.com

RETWEETS

3

LIKES

15

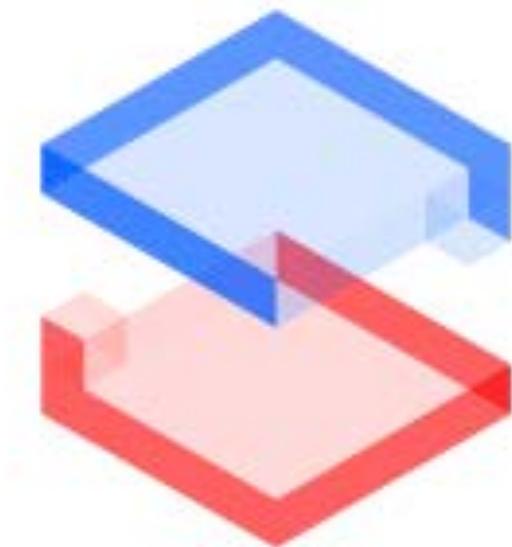


7:02 AM - 4 Nov 2016



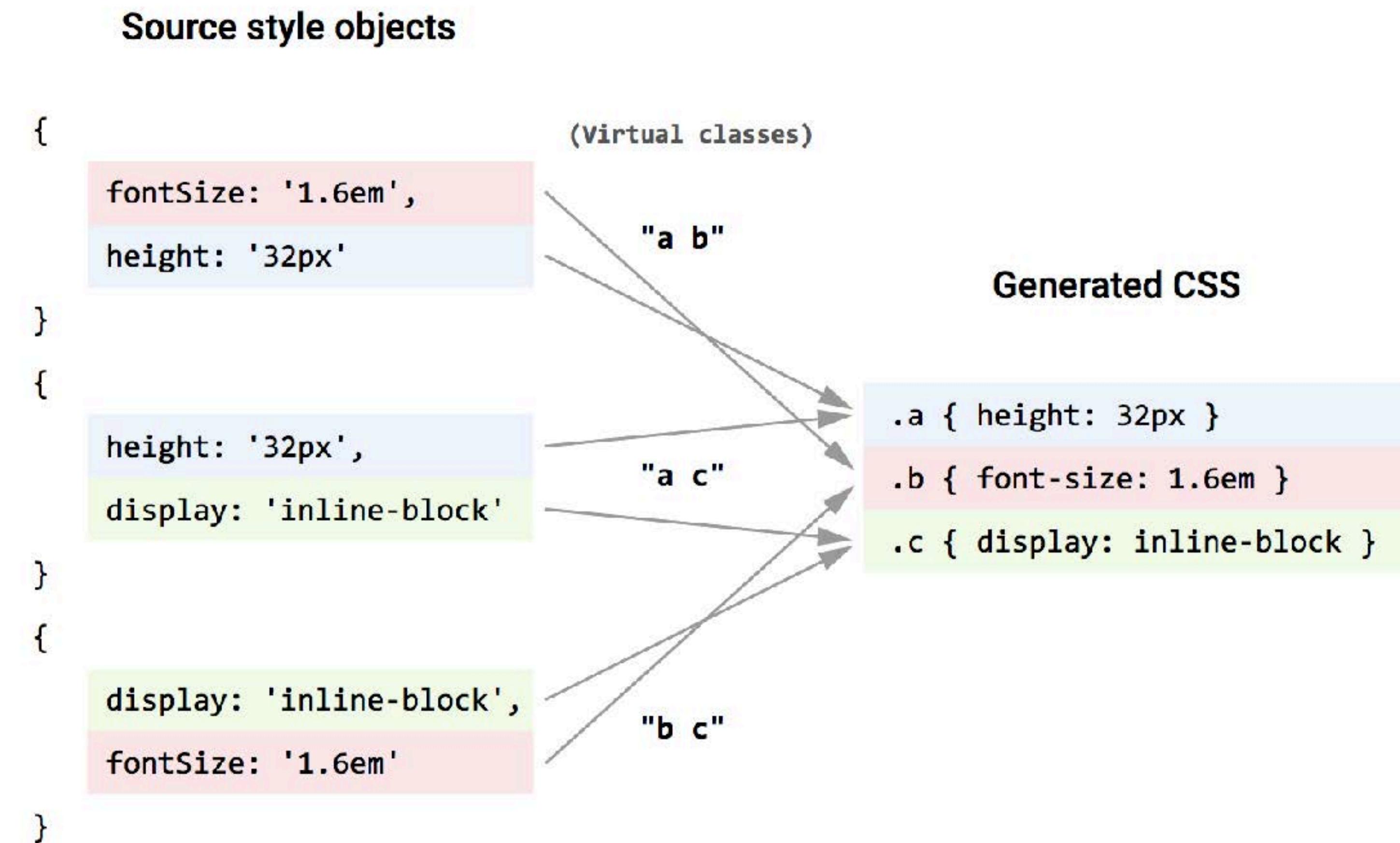
WORK HARD

Manual work



styletron

Enables CSS optimisations



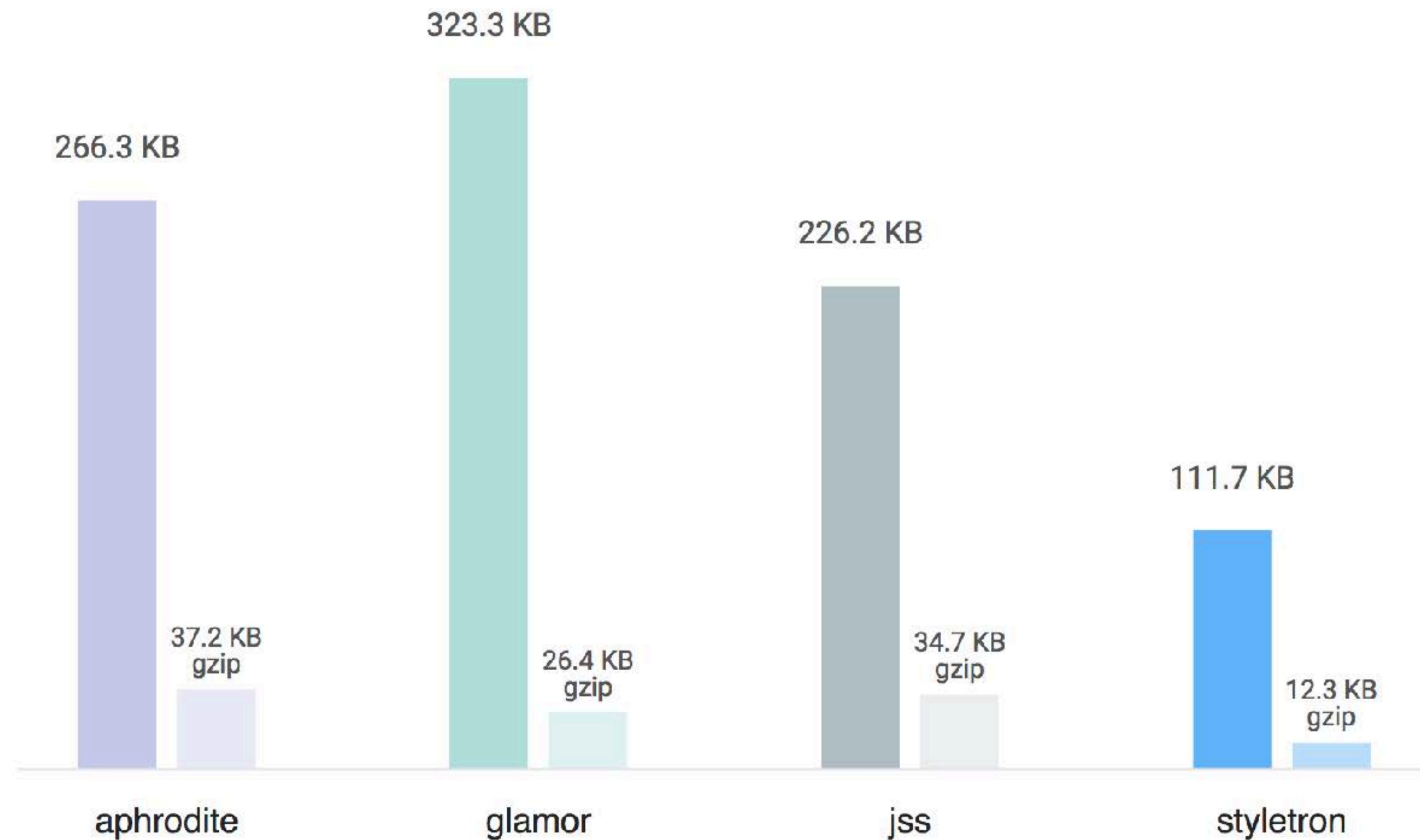
but still very easy to use API

```
import {styled} from 'styletron-react';

const Panel = styled('div', (props) => ({
  backgroundColor: props.alert ? 'orange' : 'lightblue',
  fontSize: '12px'
}));

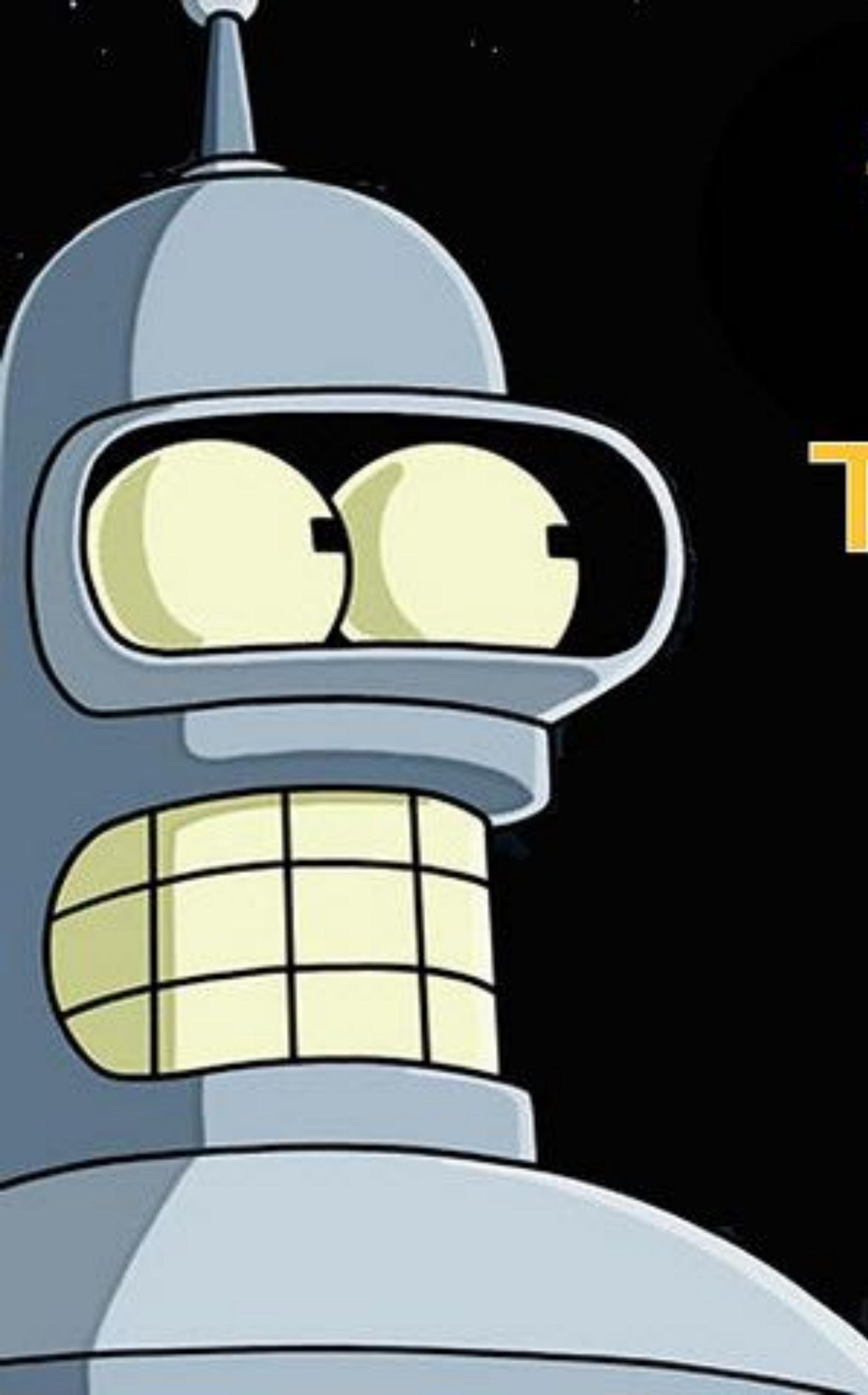
<Panel alert>Danger!</Panel>
```

airbnb.com CSS output size





Use Generate functional CSS



Kill All Humans
For A Better
TOMORROW

Bender B. Rodríguez

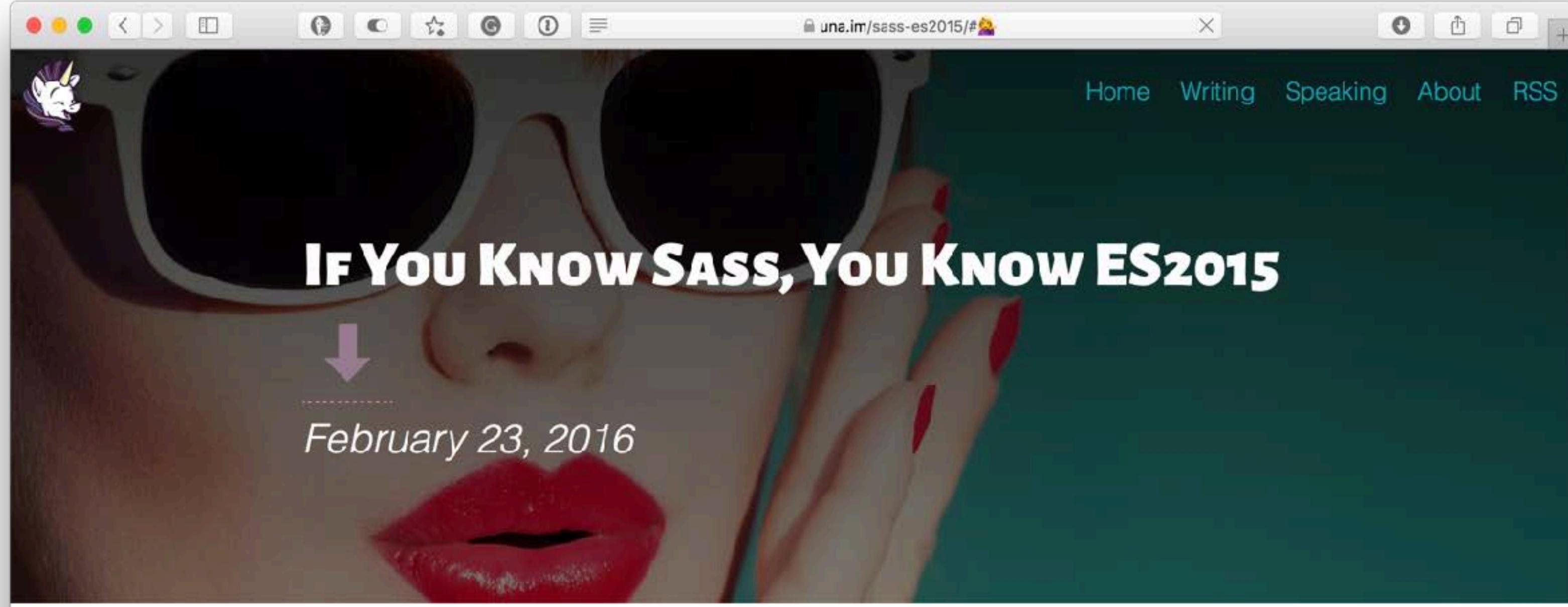


SASS has variables, mixins, etc.



JavaScript ~~SASS~~ has variables, ~~mixins~~ functions





una.im/sass-es2015/#🦄

Home Writing Speaking About RSS

IF YOU KNOW SASS, YOU KNOW ES2015

February 23, 2016

Listen to this blog post: 

If You Know Sass, You know ES2015!...well not exactly, but there [are](#) a surprising number of similarities! While learning about the [latest ECMAScript specification](#), ES2015 (formerly known as ES6), I came upon a few features that I immediately recognized from Sass. And I kept seeing more and more similarities pop up. A lot of the concepts of Sass development transfer over quite nicely, so if you know some Sass, you're probably a lot better off than you may think when it comes to diving into modern JavaScript, and visa versa.

<https://una.im/sass-es2015/#🦄>

Variables

Sass

```
$color: "red";
```

JS

```
let color = "red";
```

Lists

Sass

```
$colorList: "red", "yellow", "green";  
@each $color in $colorList {  
  @debug $color;  
}
```

JS

```
const colorArray = ["red", "yellow", "green"];  
for (const color of colorArray) {  
  console.log(color);  
}
```

Functions



```
@mixin funCircle($size, $gradient ... ) {  
  width: $size;  
  height: $size;  
  border-radius: 50%;  
  background: radial-gradient($gradient);  
}  
  
.entrancing {  
  @include funCircle(50px, blue 10%, red 80% ,pink);  
}
```



```
function PrintMe(firstArg, ... theRest) {  
  console.log(firstArg, theRest);  
}  
  
PrintMe('Hello', 'You', 'Look', 'Nice');
```

`${ }`

polished

<https://polished.js.org/>

```
// Styles as object usage
const styles = {
  background: lighten(0.2, '#CCCD64'),
  background: lighten(0.2, 'rgba(204,205,100,0.7)'),
}

// styled-components usage
const div = styled.div`  

  background: ${lighten(0.2, '#FFCD64')};  

  background: ${lighten(0.2, 'rgba(204,205,100,0.7)')};  

`  

// Output  

element {
  background: "#e5e6b1";
  background: "rgba(229,230,177,0.7)";
}
```

Just use the platform*!

* Or, just Google Chrome?

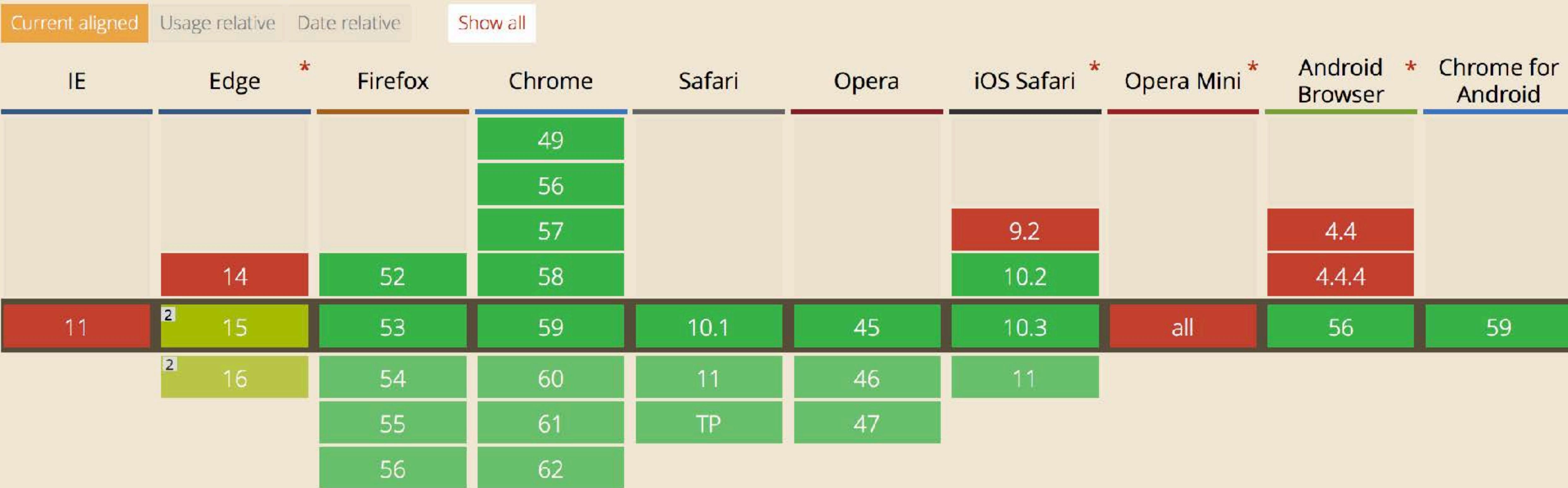
CSS Variables (Custom Properties)

CR

Global

71.49%

Permits the declaration and usage of cascading variables in stylesheets.



Shadow DOM v0

WD

Global

70.1%

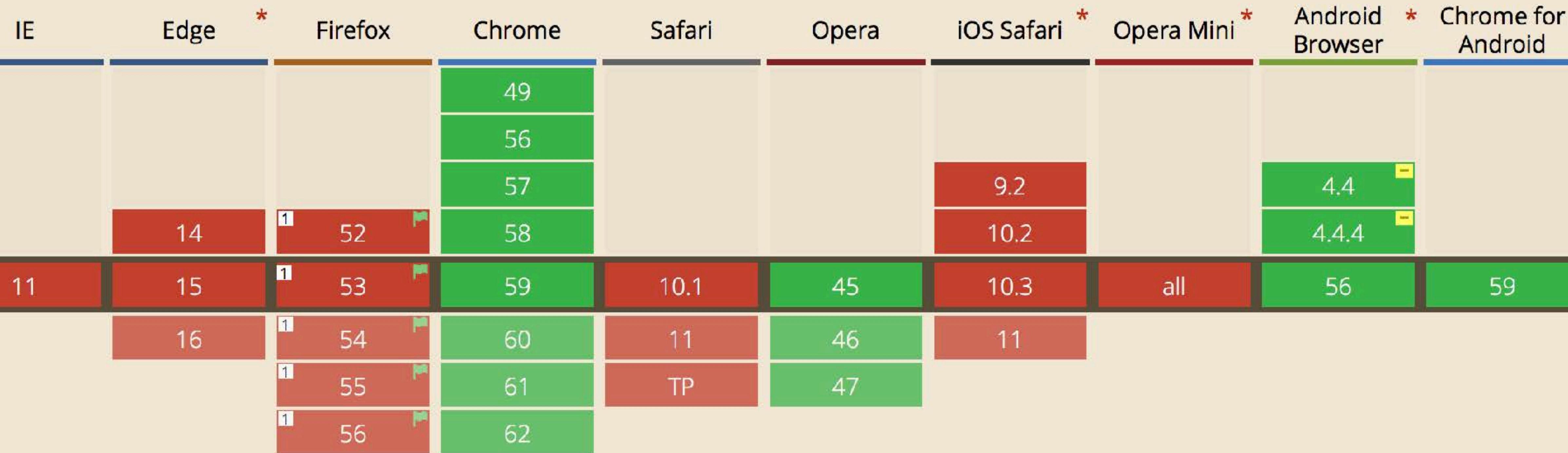
unprefixed:

64.52%

Method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM.

Current aligned Usage relative Date relative

Show all



CSS @apply rule

📄 - UNOFF

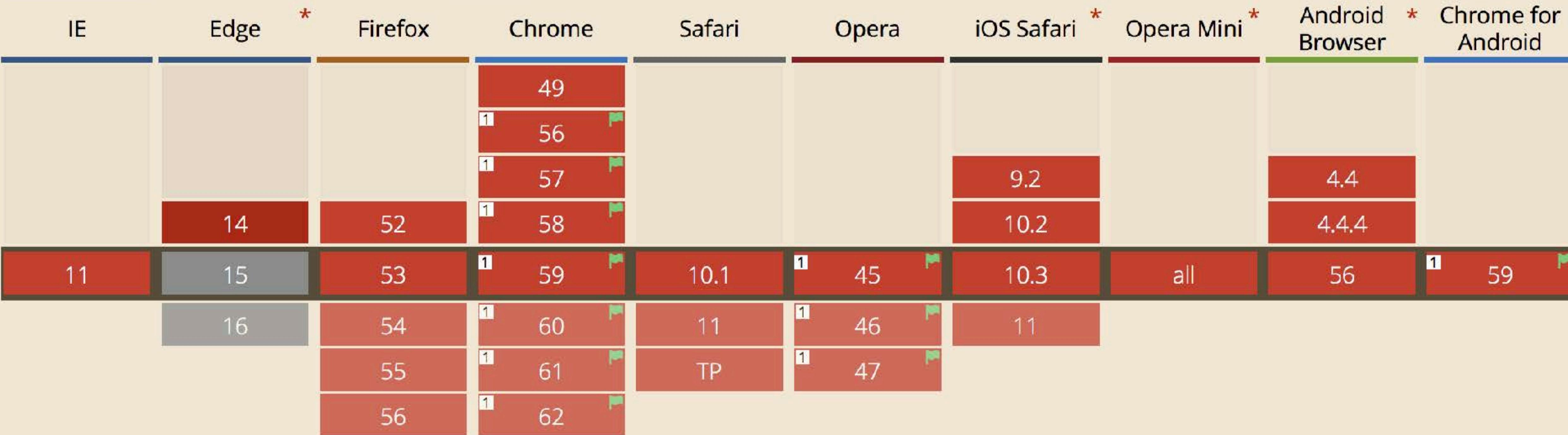
Global

0%

Allows a set of CSS properties to be applied using a named variable

Current aligned Usage relative Date relative

Show all



glam

stability experimental

```
npm install glam
```

- super fast + small (<2k gz)
- create real css files
- future facing css : nested selectors, autoprefixing, etc
- real dynamic props / `@apply` !
- Make Alex Happy (tm)

plugin options

glam can 'polyfill' for browsers that don't support css vars and / or `@apply`. fancy! you could then generate separate bundles targeting different browsers.

styled-jsx

```
export default () => (
  <div>
    <p>only this paragraph will get the style :)</p>
    { /* you can include <Component />s here that include
       other <p>s that don't get unexpected styles! */ }
    <style jsx>{
      p {
        color: red;
      }
    `}</style>
  </div>
)
```

styled-jsx

```
import _JSXStyle from 'styled-jsx/style'

export default () => (
  <div data-jsx='cn2o3j'>
    <p data-jsx='cn2o3j'>only this paragraph will get the style :)</p>
    <_JSXStyle styleId='cn2o3j' css={`p[data-jsx=cn2o3j] {color: red;}`} />
  </div>
)
```

Features

- Full CSS support, no tradeoffs in power
- Runtime size of just **2kb** (gzipped, from 6kb)
- Complete isolation: Selectors, animations, keyframes
- Built-in CSS vendor prefixing
- Very fast, minimal and efficient transpilation (see below)
- High-performance runtime-CSS-injection when not server-rendering
- Future-proof: Equivalent to server-renderable "Shadow CSS"
- Works like the deprecated `<style scoped>`, but the styles get injected only once per component

To complicated to setup



*How many of you know how to extract
critical CSS or doing that?*

Critical CSS with <>

```
import { renderToString } from 'react-dom/server'
import { ServerStyleSheet } from 'styled-components'

const sheet = new ServerStyleSheet()
const html = renderToString(sheet.collectStyles(<YourApp />))
const css = sheet.getStyleTags()
```

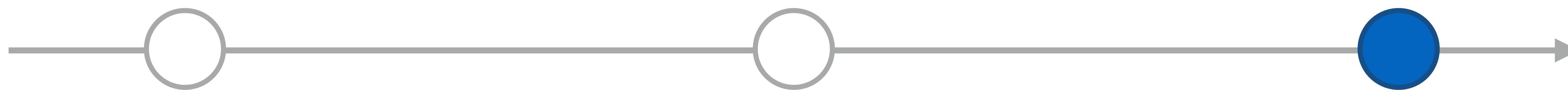
“If you build your app with progressive enhancement in mind, despite being written entirely in JavaScript, it might not require JavaScript on the client at all.”

<https://medium.com/seek-blog/a-unified-styling-language-d0c208de2660>

Past

Current time

Future



Universal rendering



```
import React from 'react';
import {AppRegistry, Pano, Text, View} from 'react-vr';

class WelcomeToVR extends React.Component {
  render() {
    // Displays "hello" text on top of a loaded 360 panorama image.
    // Text is 0.8 meters in size and is centered three meters in front of you.
    return (
      <View>
        <Pano source={asset('chess-world.jpg')} />
        <Text
          style={{
            fontSize: 0.8,
            layoutOrigin: [0.5, 0.5],
            transform: [{translate: [0, 0, -3]}],
          }>
          hello
        </Text>
      </View>
    );
  }
};

AppRegistry.registerComponent('WelcomeToVR', () => WelcomeToVR);
```

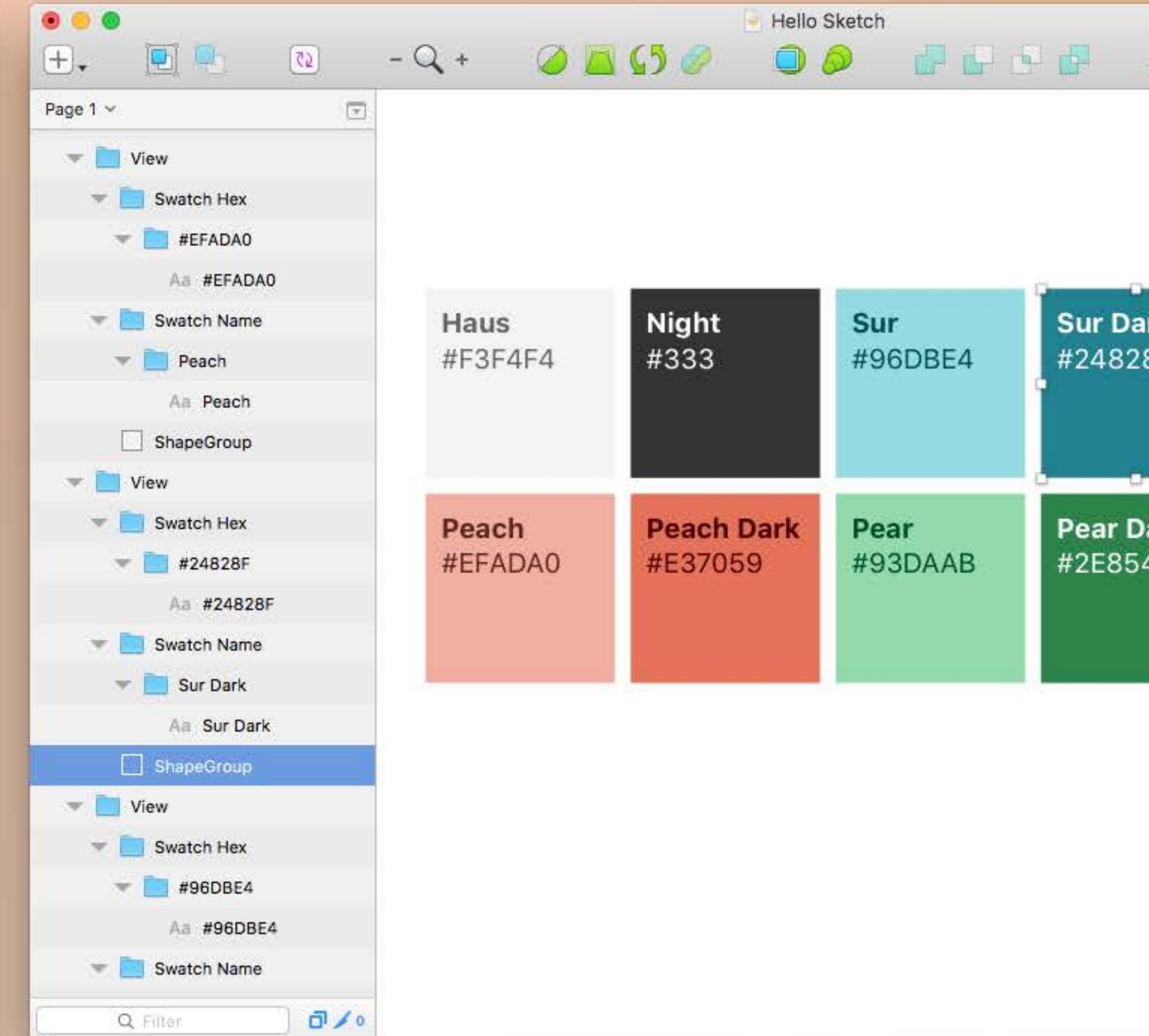


React Sketch.app

```
my-command.js — ~/code/react-sketchapp/heilo-sketch
my-command.js

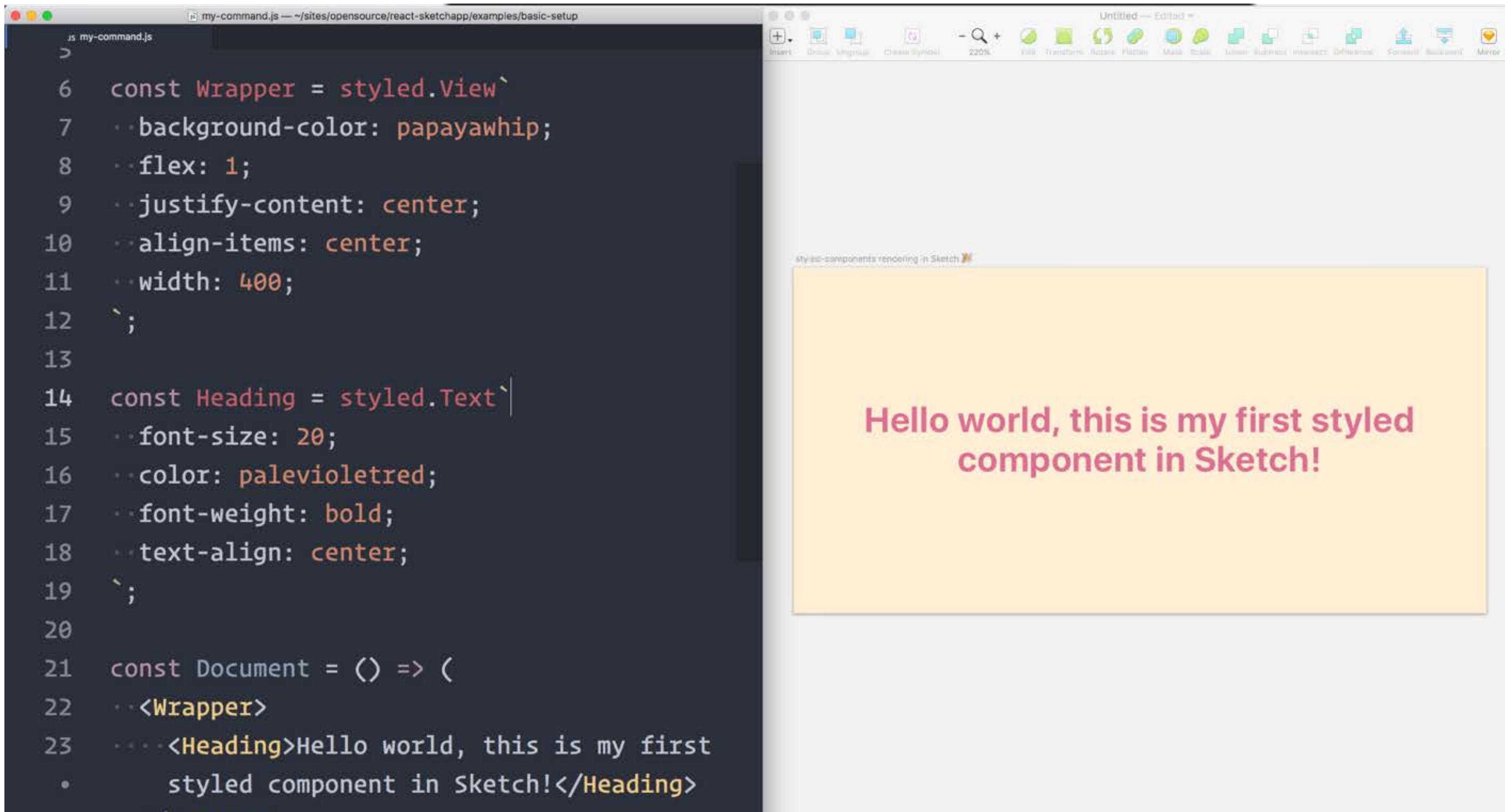
28   {name}
29     </Text>
30     <Text name="Swatch Hex" style={{ color: textColor(hex) }}>
31       {hex}
32     </Text>
33   </View>
34 );
35
36 const Document = ({ colors }) => (
37   <View style={styles.wrapper}>
38     {Object.keys(colors).map(color => <Swatch name={color} hex={colors[co
39     </View>
40 );
41
42 const colorList = {
43   Haus: '#F3F4F4',
44   Night: '#333',
45   Sur: '#96DBE4',
46   'Sur Dark': '#24828F',
47   Peach: '#EFADAO',
48   'Peach Dark': '#E37059',
49   Pear: '#93DAAB',
50   'Pear Dark': '#2E854B',
51 };
52
53 export default context =>
54   render(<Document colors={colorList} />, context);
55
```

File 0 Project 0 ✓ No Issues LF I UTF-8 Babel gold--readme ↑ 1 +55 git+ 21





styled-components for Sketch!



The image shows a Mac desktop with two windows open. On the left is a code editor window titled 'my-command.js' showing the following code:

```
js my-command.js
  1
  2
  3
  4
  5
  6  const Wrapper = styled.View`  
  7    background-color: papayawhip;  
  8    flex: 1;  
  9    justify-content: center;  
 10   align-items: center;  
 11   width: 400;  
 12 `;  
 13  
 14  const Heading = styled.Text`  
 15    font-size: 20;  
 16    color: palevioletred;  
 17    font-weight: bold;  
 18    text-align: center;  
 19 `;  
 20  
 21  const Document = () => (  
 22    <Wrapper>  
 23      <Heading>Hello world, this is my first  
 24        styled component in Sketch!</Heading>
```

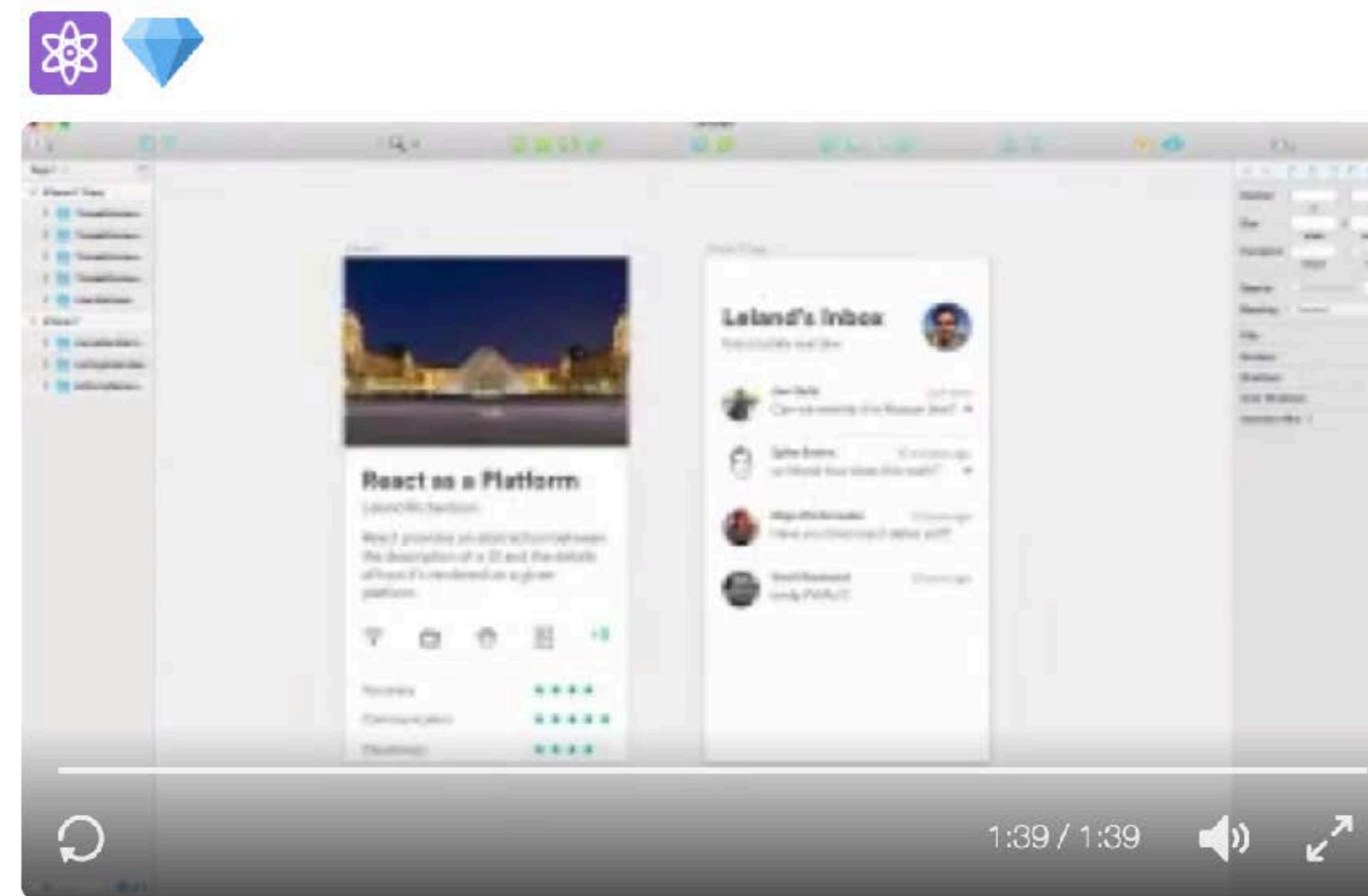
On the right is a Sketch application window titled 'Untitled — Edited —' showing a yellow rectangle with the text 'Hello world, this is my first styled component in Sketch!' in a bold, palevioletred font.



@jongold 
@jongold

Following

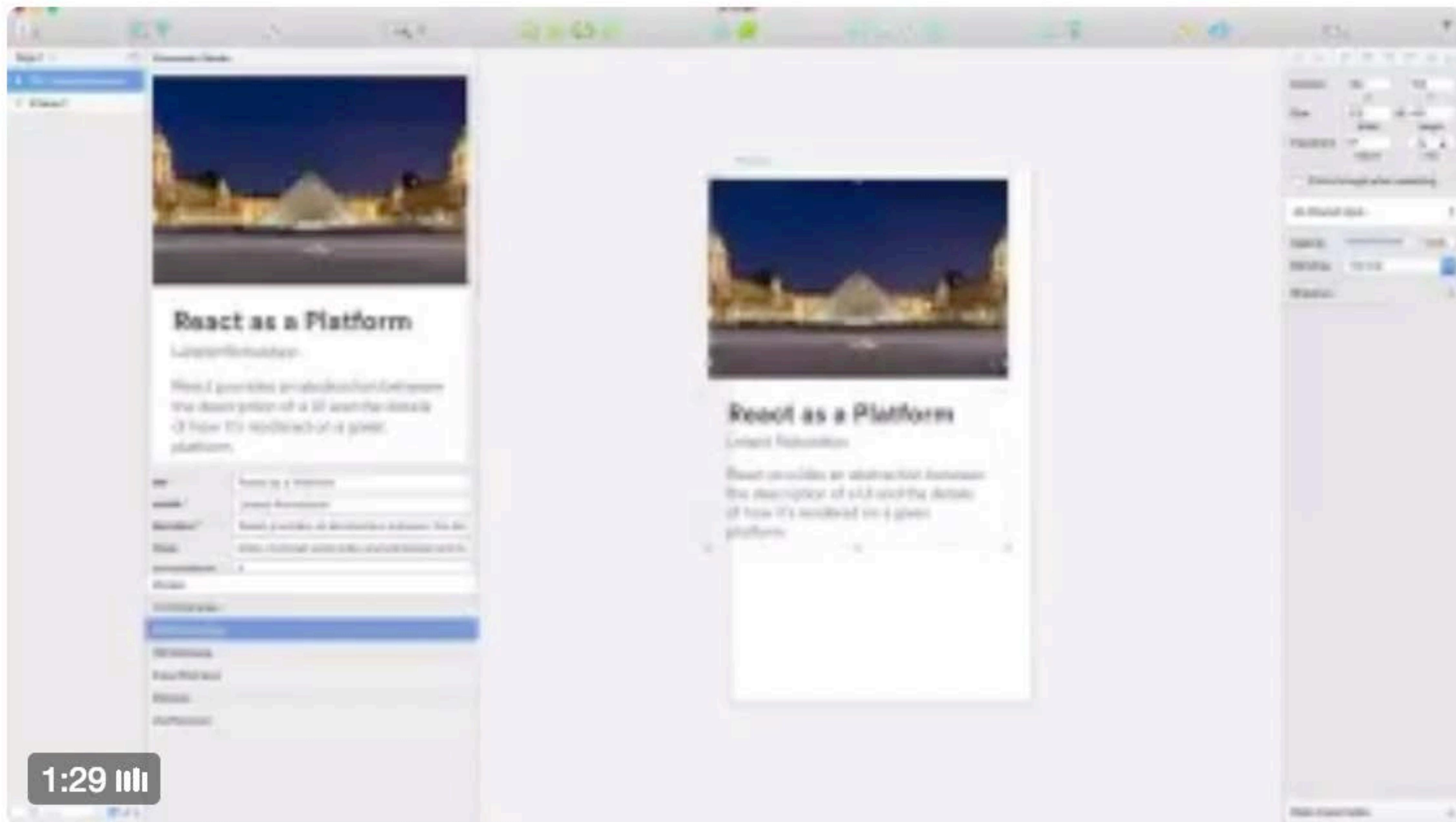
FAQ: "but why would you want to render React Native components to webviews or Sketch?"
me: "hold my kombucha"



Retweets
390

Likes
1,224





**Design systems shared between
designers and developers!**



Final thoughts



Lea Verou

@LeaVerou

Follow



CSS-in-JS is a cry for help. We at [@csswg](#) should pay attention to this and address the issues before it gets worse.

Mark Dalgleish [@markdalgleish](#)

Finally published "A Unified Styling Language" on CSS-in-JS, why people do it, and why the CSS community should care [medium.com/seek-blog/a-un...](https://medium.com/seek-blog/a-unified-styling-language-4a2a2a2a2a2a)

Retweets

123

Likes

282



1:37 PM - 24 May 2017 from [Frankfurt on the Main, Germany](#)

24

123

282



“CSS-in-JS enforces the best practices through technology and shared knowledge.”



Thank you!



Andrey Okonetchnikov

UI Engineer @ Feedly

[@okonetchnikov](https://twitter.com/okonetchnikov)

<http://okonet.ru>

<https://github.com/okonet>