STYLE GUIDE DRIVEN DEVELOPMENT
with REACT AND CSS MODULES

JENNIFER STERN

LEAD UI ENGINEER @ INSTRUCTURE

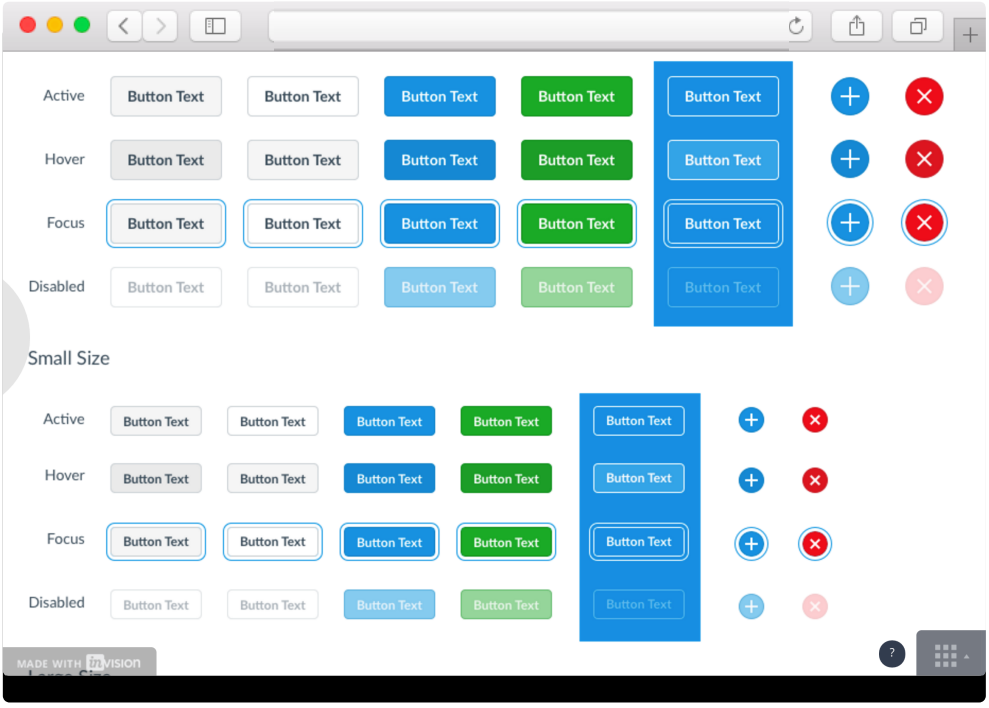
jstern@instructure.com

WHAT'S A STYLE GUIDE?

> A style guide is a set of standards for the writing and design of documents, either for general use or for a specific publication, organization, or field....
>
> **A STYLE GUIDE ESTABLISHES AND ENFORCES STYLE TO IMPROVE COMMUNICATION.**

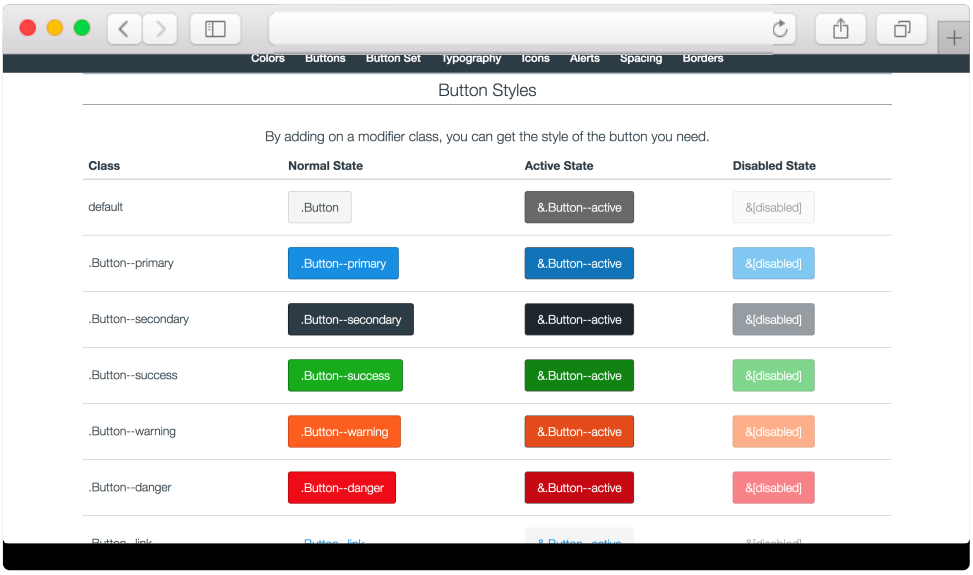
- Wikipedia

STATIC STYLE GUIDE



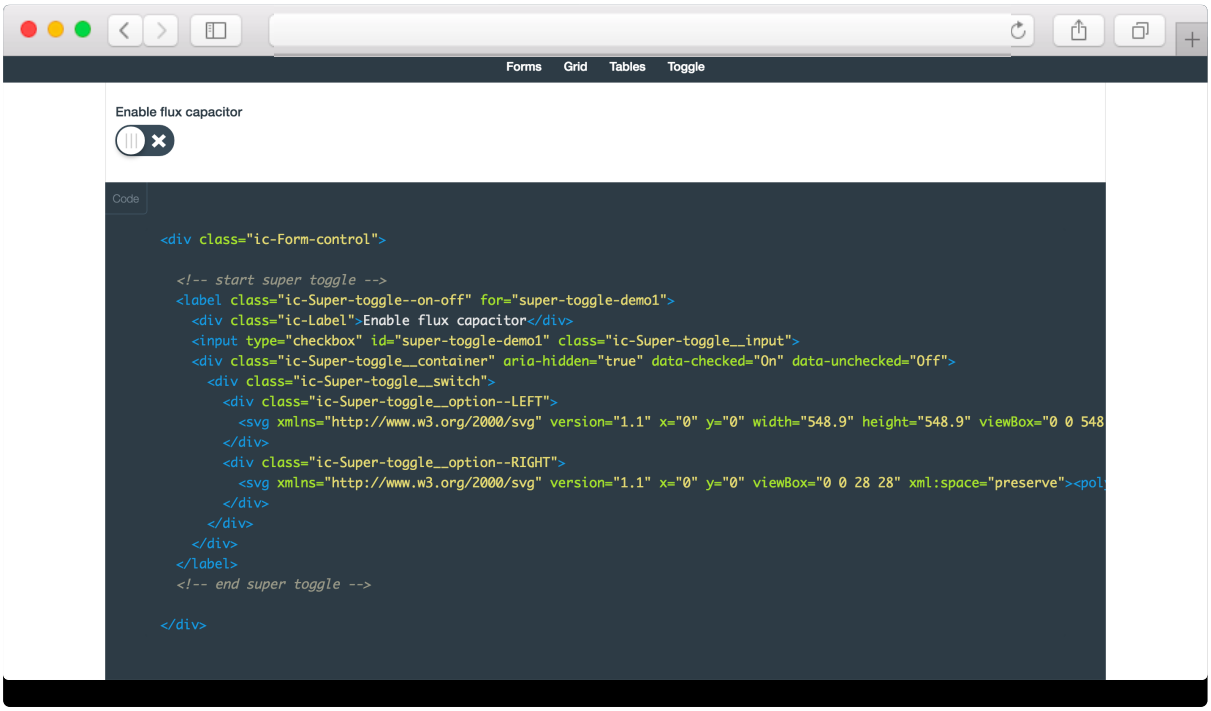
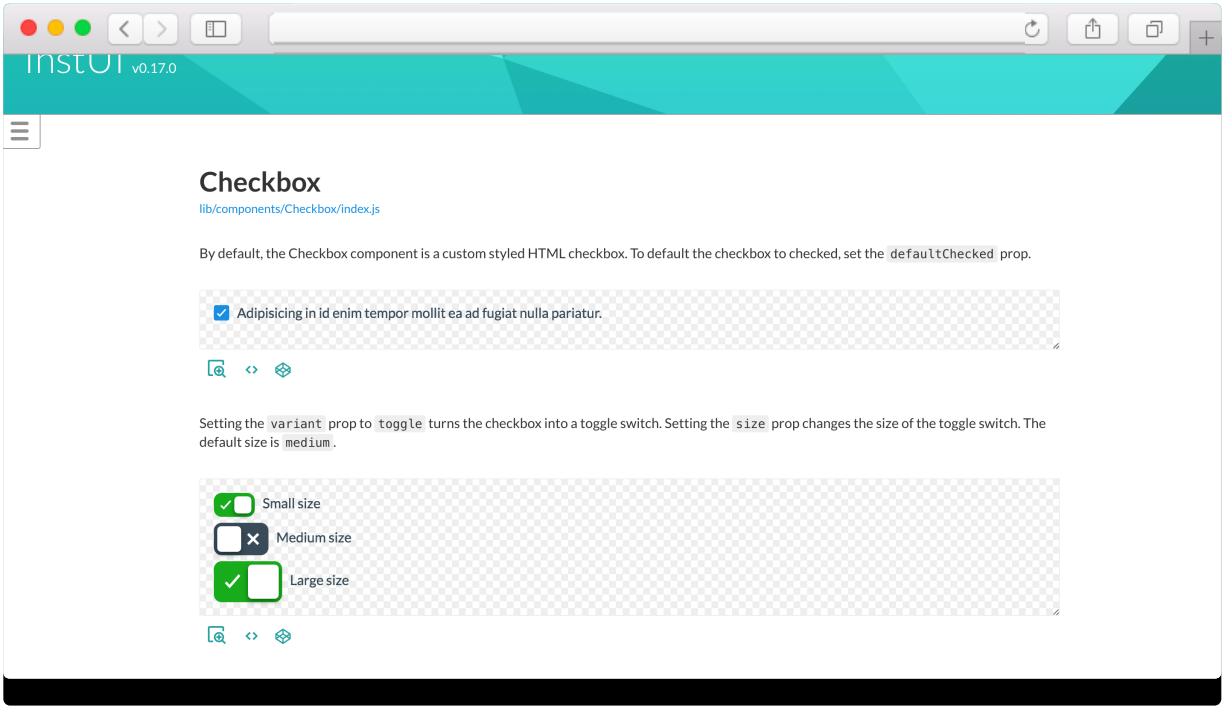
LIVE STYLE GUIDE

(or pattern library)



IN A LIVE STYLE GUIDE

DOCUMENTATION LIVES WITH THE SOURCE CODE IN THE SAME REPOSITORY

```
  3. //
-------
  4.
  5.
  6. // Base styles
  7. // ----------------------------------------
-------
  8.
  9. // Core
 10.
 11. /*
 12. @styleguide Buttons
 13.
 14. ## The Basic Button
 15.
 16. ```html
 17. <button class="Button" type="button">My
Awesome Button</button>
 18. ```
 19. ## Button Styles
 20.
 21. By adding on a modifier class, you can get the
```

A CUSTOM CHECKBOX IN **SASS**



A CUSTOM CHECKBOX IN **REACT**



CSS, JS, HTML & DOCUMENTATION

IN ONE PLACE

```
 16. import theme from '../theme.js'
 17.
 18. /**
 19.   By default, the Checkbox component is a
custom styled HTML checkbox. To default the
checkbox to checked,
 20.   set the `defaultChecked` prop.
 21.
 22.   ```jsx_example
 23.     <Checkbox label={lorem.sentence()}
value="medium" defaultChecked />
 24.   ```
 25.
 26.   Setting the `variant` prop to `toggle` turns
the checkbox into a toggle switch.
 27.   Setting the `size` prop changes the size of
the toggle switch. The default size
 28.   is `medium`.
 29.
 30.   ```jsx_example
 31.   <div>
 32.     <Checkbox label="Small size" value="small"
```

> Make the right things easy and the wrong things hard.

- Jonathan Snook

And now I'll attempt to live code...

> CSS in JS?
>
> 1. Global Namespaces
> 2. Dependencies
> 3. Dead Code Elimination
> 4. Minification
> 5. Sharing Constants
> 6. Non-Deterministic Resolution
> 7. Isolation

- Christopher "vjeux" Chedeau

CSS MODULES

(ISOLATE COMPONENT STYLES **WITHOUT WRITING CSS IN JS**)

button.css

```
.root {
  all: initial; /* this prevents external CSS from breaking this component */
  appearance: none;
  box-sizing: border-box;
  display: inline-block;
  vertical-align: middle;
  position: relative;
  overflow: visible;

  margin: 0;

```

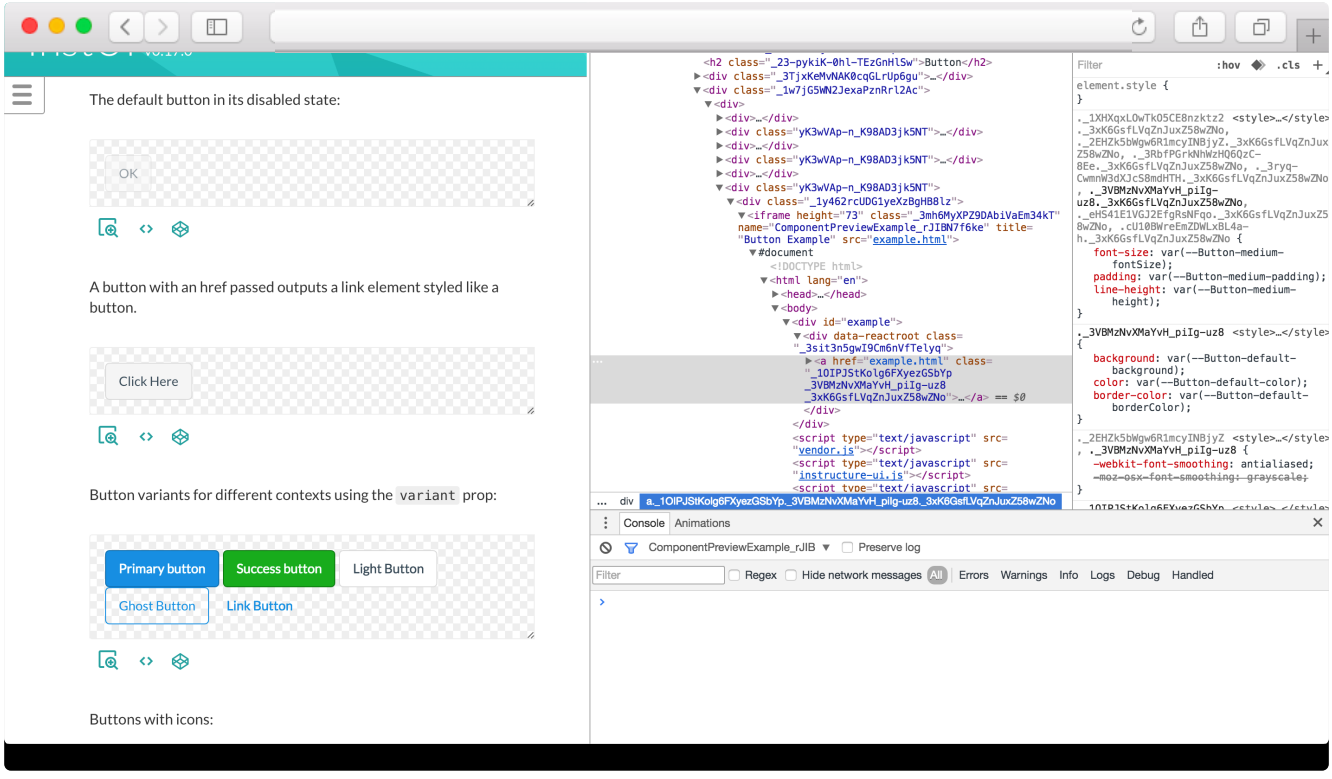
CSS MODULES

LOADING A CSS FILE IN A JS COMPONENT

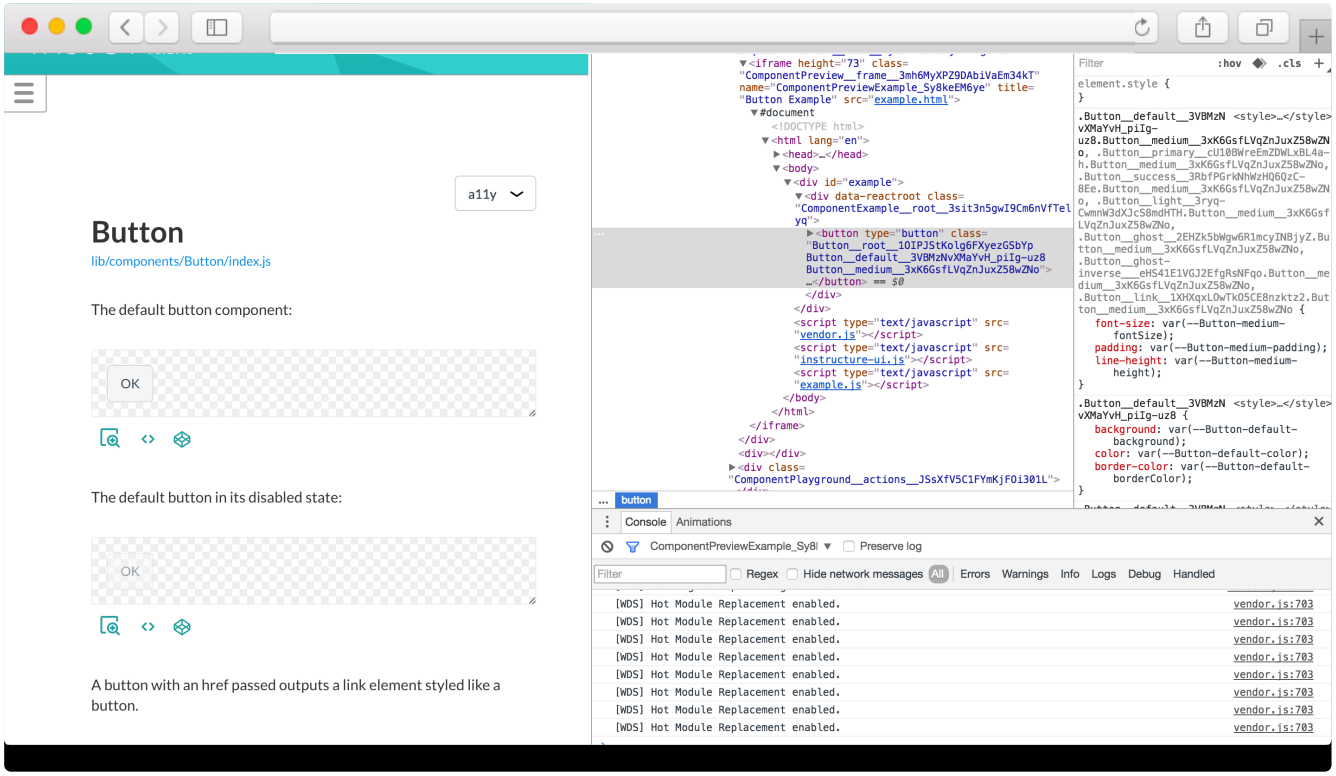
button.js

```
 1. import React, { Component, PropTypes } from
'react'
 2. import ReactDOM from 'react-dom'
 3. import classnames from 'classnames'
 4. import themeable from '../../util/themeable'
 5. import keycode from 'keycode'
 6. import { omitProps } from
'../../util/passthroughProps'
 7.
 8. import styles from './styles.css'
 9. import theme from './theme.js'
10.
```

WITH PRODUCTION CONFIG

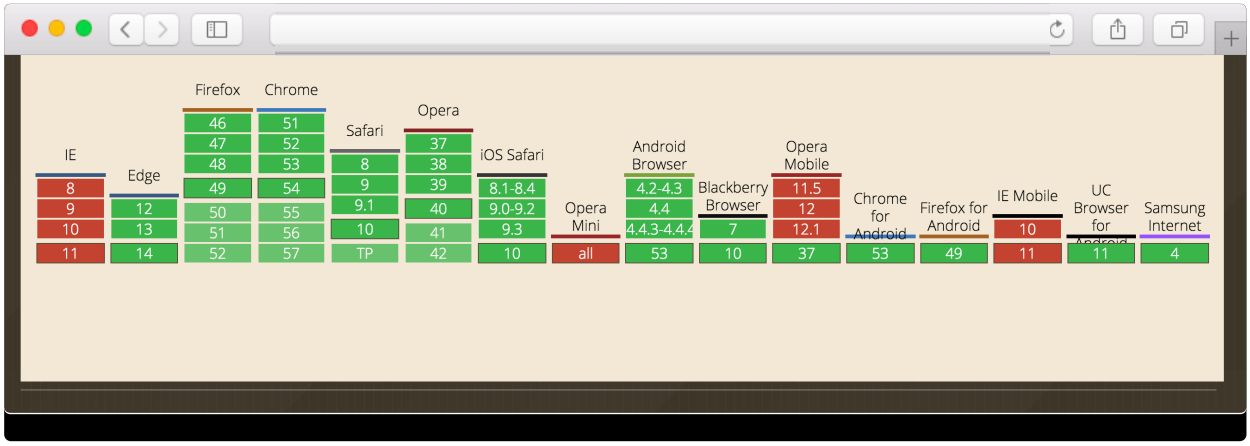


WITH DEVELOPMENT CONFIG



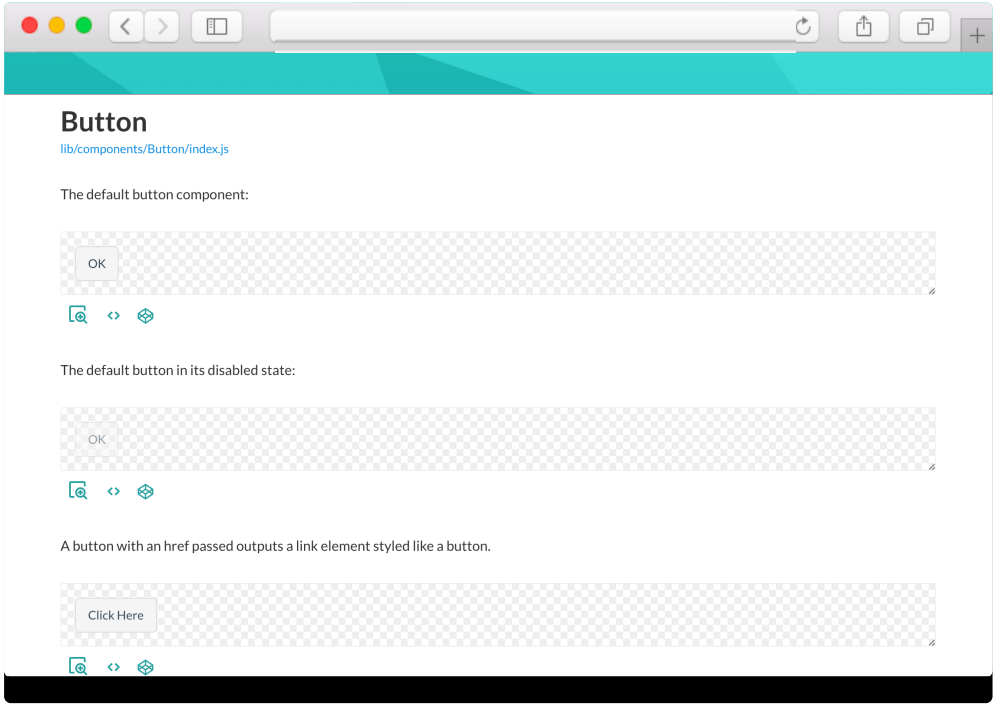
PREVENT THE CASCADE

WITH `all: initial` AND POSTCSS-INITIAL



ACCESSIBILITY

ACCESSIBLE BUTTONS



ADDING (JS) BEHAVIOR TO COMPONENTS FOR **ACCESSIBILITY**

button.js

```
 1. import React, { Component, PropTypes } from
'react'
 2. import ReactDOM from 'react-dom'
 3. import classnames from 'classnames'
 4. import themeable from '../../util/themeable'
 5. import keycode from 'keycode'
 6. import { omitProps } from
'../../util/passthroughProps'
 7.
 8. import styles from './styles.css'
 9. import theme from './theme.js'
10.
```

UNIT TESTS FOR ACCESSIBILITY USING **AXE-CORE**

checkbox.test.js

```
1. import React from 'react'
2. import Checkbox from '../index'
3.
4. describe('<Checkbox />', function () {
5.   const testbed = createTestbed(
6.     <Checkbox label="fake label"
defaultChecked value="someValue" name="someName" />
7.   )
8.
9.   it('renders an input with type "checkbox"',
function () {
10.     const subject = testbed.render()
```

TESTING **COLOR CONTRAST** FOR A11Y

theme.test.js

```
1. import theme from '../theme'
2. import Brands from '../../../theme/brands'
3. import { contrast } from '../../../util/color'
4.
5. describe('Button.theme', function () {
6.   describe('with the default theme', function
() {
7.     const variables = theme(Brands.inst)
8.
9.     describe('default', function () {
10.       it('should have a background and text
colors that meet 3:1 contrast', function () {
```

THEMES

SASS VARIABLES **IN THE MONOLITH**

variables.scss

```
1. /// TOC
2.
//=============================================================
=============================================
3. //
4. // - Canvas Brand Variants
5. // - Custom Branding
6. // - Canvas LMS Color Sheet
7. // - Canvas Theme Color Variables
8. //
9.
//=============================================================
=============================================
```

GLOBAL BRAND VARIABLES DEFINED IN JS

brand.js

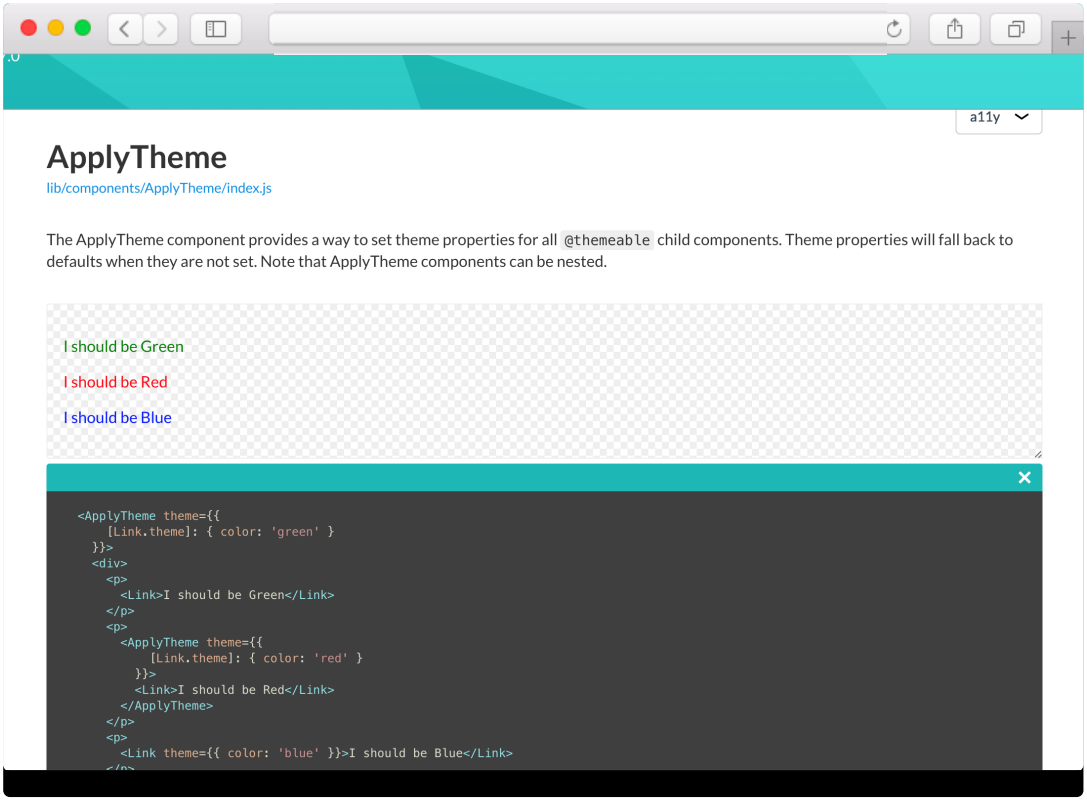
```
export default Object.freeze({
  // These need to meet 3:1 contrast requirements when used as text against background
  // colors lightest or when used as a background with text colors lightest
  brand: '#008EE2',
  info: '#008EE2',
  success: '#00AC18',
  action: '#BF32A4',
  danger: '#EE0612',
  warning: '#FC5E13',

```

COMPONENT VARIABLES DEFINED IN JS

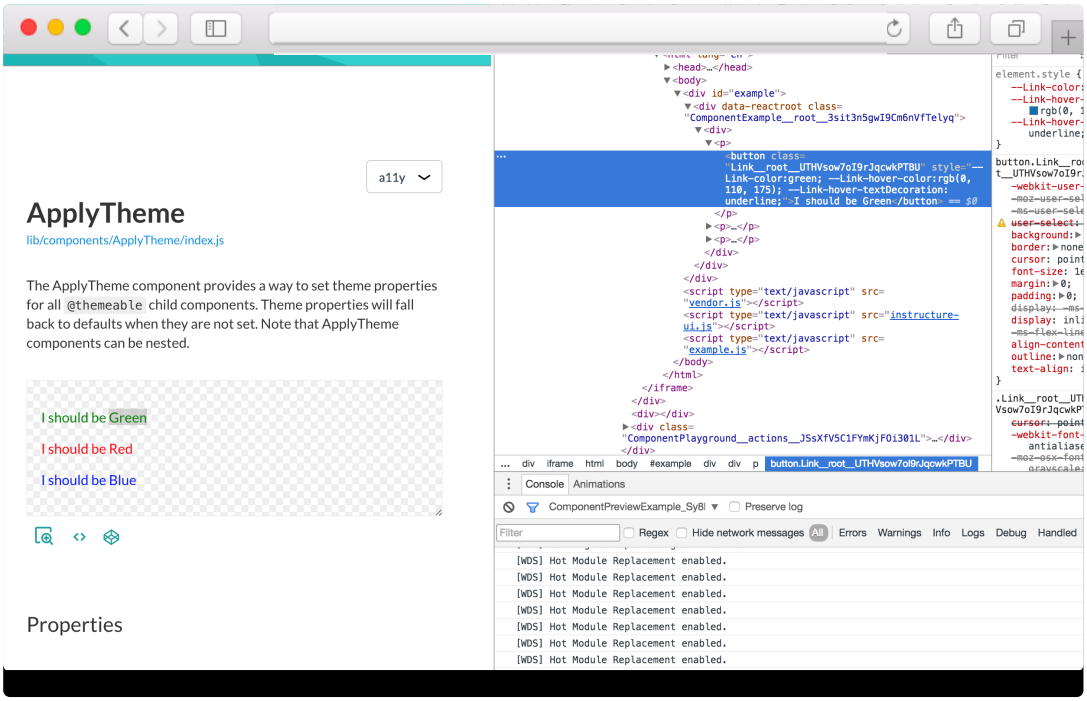
theme.js

```
1. import { darken } from '../../util/color'
2.
3. export default function generator ({ colors,
typography }) {
4.   return {
5.     fontFamily: typography.fontFamily,
6.     fontWeight: typography.fontWeightNormal,
7.
8.     color: colors.brand,
9.     textDecoration: 'none',
10.    hover: {
11.      color: darken(colors.brand, 10),
```

APPLYING THEMES AT RUN TIME



CSS CUSTOM PROPERTIES

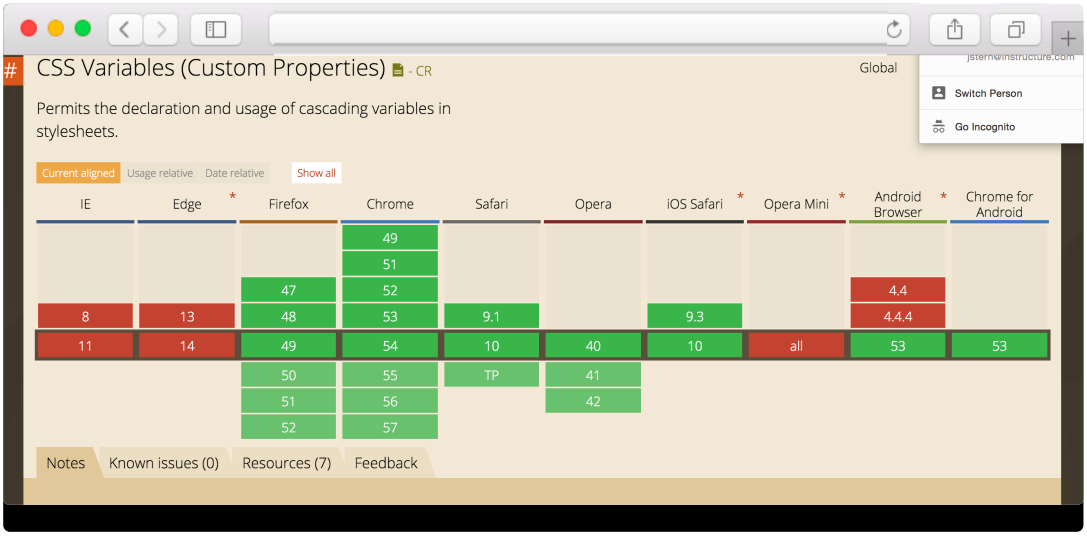


CSSStyleDeclaration.setProperty()

```
const setNodeProperties = function (node, prop
  Object.keys(properties).forEach(function (pr
    const value = properties[propertyName]
    if (value) {
      node.style.setProperty(propertyName, valu
    }
  })
}
```

CSS CUSTOM PROPERTIES

(we wrote a polyfill for IE)



BUILT TO SCALE

We're using our UI components in multiple applications as we start breaking up the monolith into microservices

&

Our professional services team is using the library to build custom integrations with a seamless UX.

QUESTIONS?

https://github.com/instructure/instructure-ui

WE'RE HIRING!

Contact Ariel: apao@instructure.com

RESOURCES AND LINKS

- react-docgen
- eslint
- stylelint
- axe-core
- eslint-plugin-jsx-a11y
- postcss-initial
- css modules
- webpack css-loader