

MERRY CHRISTMAS, YA'LL!

STYLIN' AT FAMILYSEARCH

JAKETRENT.COM Next Wed, what was that date...?

GOALS

- 1. Why multiple techniques?**
- 2. Styling opinions**
- 3. Code how you code**
- 4. Share questions and answers**

“Problems with CSS at Scale”

Christopher Chedeau

NationJS 2014

GLOBAL NAMESPACE

```
.my-selector {}  
.etc {}  
.my-selector {  
    /* accidental overrides */  
}
```

NO EXPLICIT DEPENDENCIES

```
<div className="from-somewhere" />
```

DIFFICULT TO ELIMINATE CODE

```
.who-knows {}  
.what-can {}  
.safely {}  
.be-removed {}
```

UNOPTIMAL MINIFICATION

.selector-names {}

CAN'T EASILY SHARE CONSTANTS

```
.my-selector {  
  /* same as JS */  
  animation-duration: 400ms;  
}
```


NON-DETERMINISTIC RESOLUTION

```
const link = document.createElement('link')  
link.rel = 'stylesheet'  
link.href = 'some-new-stylesheet.css'  
document.head.appendChild(link)
```

DIFFICULT TO ISOLATE

```
.some-component > div > a {  
  color: red;  
}
```

NEW PATTERNS BORN

Inline styles

CSS-in-JS

CSS modules

MY OPINIONS

“Things I like”

COLOCATION

```
<div className="selector" />
```

vs

```
<div style={{ color: red, /* etc */ }} />
```

COLOCATION

```
const MainComponent = props => {  
  return (  
    <div className="so-styled">  
      The main show {props}  
    </div>  
  )  
}
```

COLOCATION

mycomponent.tsx

mycomponent.styles.ts

COLOCATION

```
import { css } from '@emotion/react'
```

```
export const container = props => css`...`
```

```
export const button = props => css`...`
```

```
// ...
```


COLOCATION

mycomponent.tsx

mycomponent.module.css

DYNAMIC STYLES

```
<view {...withStyles(state)} />
```

#grateful

DYNAMIC STYLES

```
import styled from '@emotion/styled'
```

```
const Button = styled.button`  
  color: ${props => props.active ? 'red' : 'black'};  
`
```

DYNAMIC STYLES

```
import cx from 'classnames'

<button className={cx({
  button: true,
  'button--active': props.active
})} />
```

DYNAMIC STYLES

```
import css from './button.module.css'
```

```
<button className={  
  props.active ? css.buttonActive : css.button  
} />
```

COMPOSITION

```
.juicy {  
  color: red;  
}  
.watermelon {  
  composes: juicy;  
  background-color: green;  
}  
  
// exports  
{ watermelon: 'juicy-abc123 watermelon-abc123' }
```

LANGUAGE

CSS when I can

JS isn't bad

VARIABLE SHARING

JS

APP STYLES

APP.CSS

COMPONENT STYLES

COMPONENT.JS

STYLES in PACKAGES

```
// package.json
```

```
{  
  "style": "dist/index.css"  
}
```

SEPARATE NAMESPACE

```
// button.module.css
```

```
.button {}
```

```
.buttonActive {}
```

ISOLATED

.button {} ==> .button-**abc123** {}

CASCADE

Get to use it

Have to play nice

Defenseless

CASCADE

The good ol' days

Naive

EXPLICIT DEPENDENCIES

```
import css from './component.module.css'
```

or

```
import css from './component.module.js'
```


LIBRARIES

KISS

Stick with it

Mainstream & experiments

BEM

Lots of mileage

TALENT

Make hiring easy

TIGHTEN FEEDBACK LOOP

Fast is better

Hot reloads

SIMPLICITY

Understand

Fix

CHOICE

Make it

See it through

Create value

MASTERY

Value of experience

vs

Well-versed in opinion

YOUR CODE

CSS-in-JS

Emotion

Beautiful

Q&A TO FOLLOW INTERMINGLE

THANK YOU