

CSS



CSS meets JS

Dominic Langenegger

JS

About me

Dominic Langenegger

Studied Computer Science at ETH Zurich, Switzerland

Moved to Singapore in October 2017

Software Engineer @ Zuhlke Engineering Pte Ltd

Service provider and solution partner

1000 employees, mostly in Europe

zühlke
empowering ideas



Problems with CSS at Scale

- Global Namespace
- Dependencies
- Dead Code Elimination
- Minification
- Sharing Constants
- Non-deterministic Resolution
- Isolation

Christopher Chedeau, React Team @ facebook
<https://speakerdeck.com/vjeux/react-css-in-js>

Bootstrap



> 600 global variables

(at least they are using scss)

CSS at scale
What are our options?

Disciplined use of CSS

Naming conventions

File structure

Component hierarchy

BEM

Block, Element, Modifier

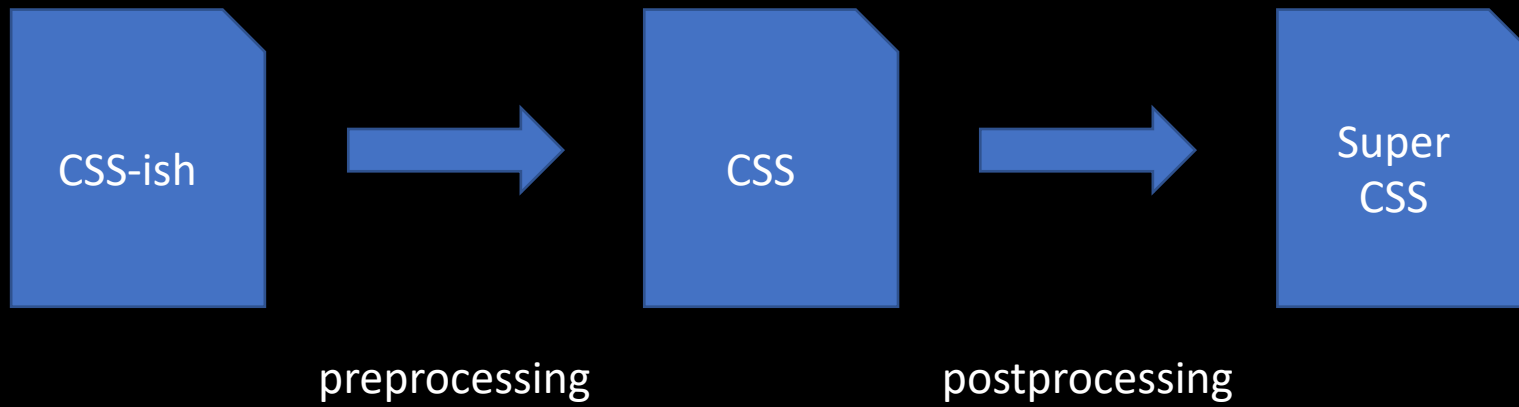
<https://en.bem.info/methodology/>

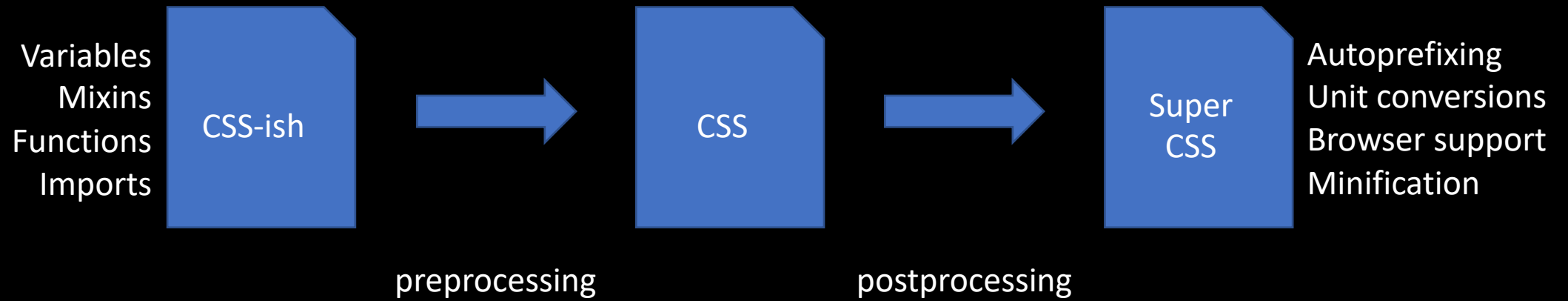
SMACSS

Scalable and Modular Architecture for CSS

<https://smacss.com/>

Can it really scale?





Let's build some buttons

Primary

Secondary

Large button

Large button

Plain old CSS *(actually Bootstrap)*

global namespace

```
.btn {  
  display: inline-block;  
  font-weight: 400;  
  color: #212529;  
  text-align: center;  
  vertical-align: middle;  
  -webkit-user-select: none;  
  -moz-user-select: none;  
  -ms-user-select: none;  
  user-select: none;  
  background-color: transparent;  
  border: 1px solid transparent;  
  padding: 0.375rem 0.75rem;  
  font-size: 1rem;  
  line-height: 1.5;  
  border-radius: 0.25rem;  
}
```

vendor prefixes

```
.btn-primary {  
  color: #fff;  
  background-color: #007bff;  
  border-color: #007bff;  
}
```

constants

```
.btn-secondary {  
  color: #fff;  
  background-color: #6c757d;  
  border-color: #6c757d;  
}
```

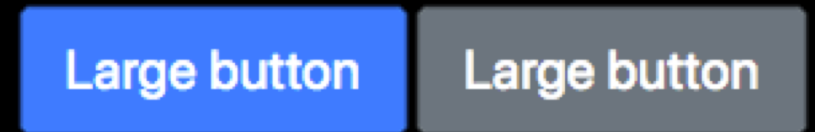
```
.btn-lg {  
  padding: 0.5rem 1rem;  
  font-size: 1.25rem;  
  line-height: 1.5;  
  border-radius: 0.3rem;  
}
```

overriding

Bootstrap



```
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</button>
```



```
<button type="button" class="btn btn-primary btn-lg">Large button</button>  
<button type="button" class="btn btn-secondary btn-lg">Large button</button>
```

SCSS

```
still global namespace...
.btn {
  display: inline-block;
  font-weight: $btn-font-weight;
  color: $body-color;
  text-align: center;
  vertical-align: middle;
  user-select: none;
  background-color: transparent;
  border: $btn-border-width solid transparent;
  @include button-size(
    $btn-padding-y,
    $btn-padding-x,
    $btn-font-size,
    $btn-line-height,
    $btn-border-radius
  );
}
```

```
.btn-lg {
  @include button-size(
    $padding-y-lg, $padding-x-lg,
    $font-size-lg, $line-height-lg,
    $border-radius-lg
  );
}

@mixin button-size(
  $padding-y, $padding-x, $font-size,
  $line-height, $border-radius
) {
  padding: $padding-y $padding-x;
  font-size: $font-size;
  line-height: $line-height;
  @include border-radius($border-radius, 0);
}
```

CSS modules

- One file per component
- Scoped locally by default
- Class names are auto-generated by loader
- Works with pre- and postprocessors

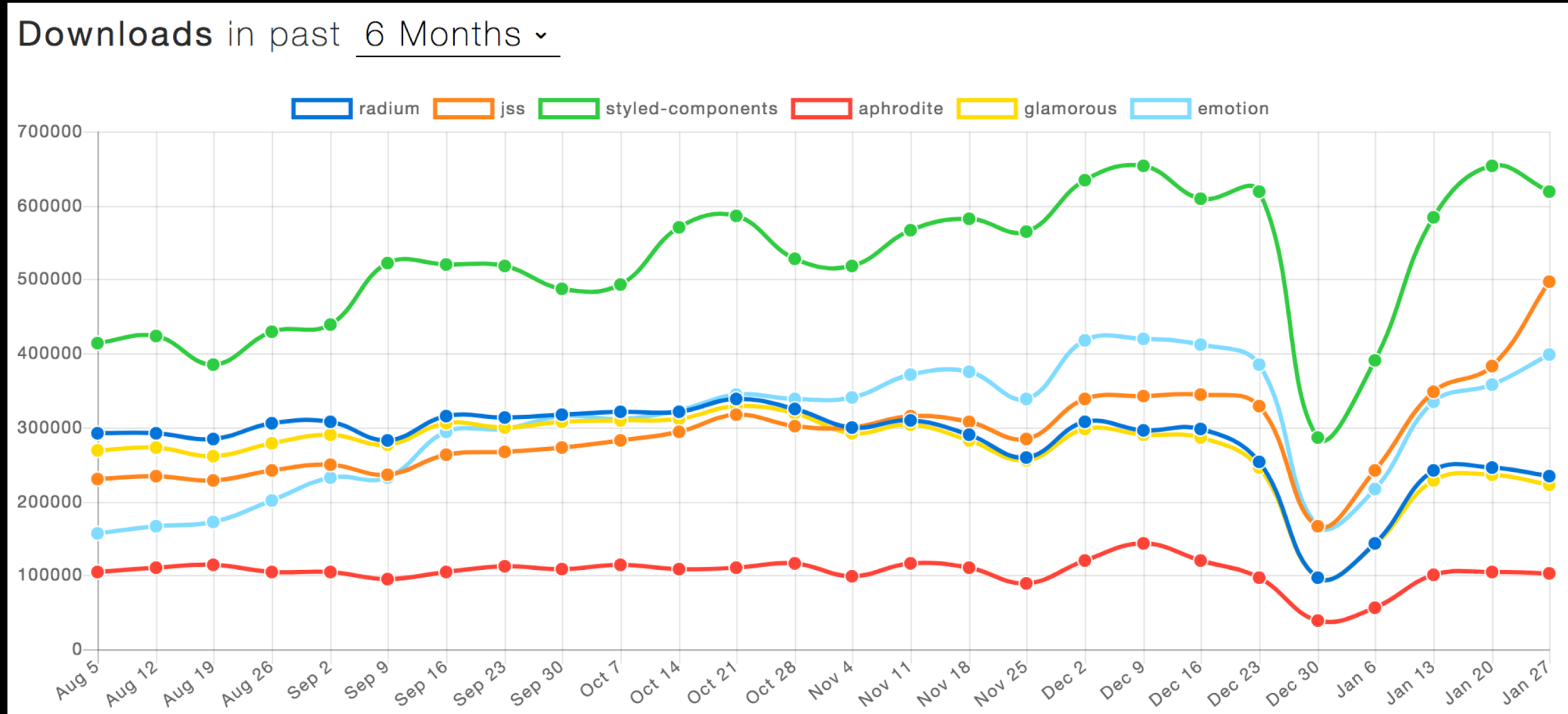
Common with React and Angular

```
import React from "react";
import style from "./button.css";

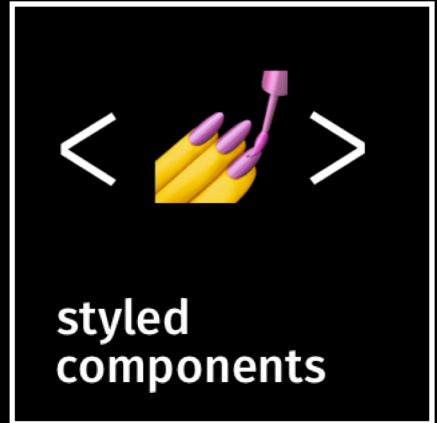
// some component
const classes =
  style.button + " " + style.primary;
return (
  <button className={classes}>
    Primary Button
  </button>
);
```

Why CSS in JS?

Framework Comparison



styled-components (React only)



```
const Button = styled.button`      uses ES6 template literal
  display: inline-block;
  background-color: ${props =>
    props.primary ? colors.primary : colors.secondary};  has access to props
  border: solid 1px;
  border-color: ${props => (props.primary ? colors.primary : colors.secondary)};
  border-radius: ${props => (props.large ? "0.3rem" : "0.25rem")};
  font-size: ${props => (props.large ? "1.25rem" : "1rem")};
  font-weight: 400;
  line-height: 1.5;                full JS capabilities
  padding: ${props => (props.large ? "0.5rem 1rem" : "0.375rem 0.75rem")};
  user-select: none;
  text-align: center;
  vertical-align: middle;
`;
```

Syntax highlighting and autocomplete in VS Code

```
<Button primary>Primary Button</Button>
```


JSS



```
const styles = {
  button: {
    display: "inline-block",
    border: `solid 1px
      ${colors.secondary}`,
    fontSize: "1em",
    padding: "0.375rem 0.75rem"
    // ...
  },
  primaryButton: {
    extend: "button",
    borderColor: colors.primary,
    backgroundColor: colors.primary
  }
};
```

```
const { classes } =
jss.createStyleSheet().attach();
styles
document.body.innerHTML = `
  <button class="${classes.primaryButton}">
    Primary Button
  </button>
  <button class="${classes.button}">
    Secondary Button
  </button>
`;
```

Emotion

- Works well with react but is framework-agnostic
- styled-component compatible API
- Faster and lighter
- Can extract static css file



CSS in JS

Pros

- Full power of JavaScript
- Completely deterministic
- Can do heavy optimization

Cons

- Full power of JavaScript
- Good design is optional
- Sometimes unusual syntax
- Less editor/tooling support
- Some frameworks don't support all of CSS

Even with CSS in JS,
you still need to know about CSS

Things to look out for

- Syntax and Editor
- CSS Feature Support
 - vendor prefixing
 - pseudo-classes
 - media queries
- SSR
- Static file generation
- Performance
 - Speed
 - Size

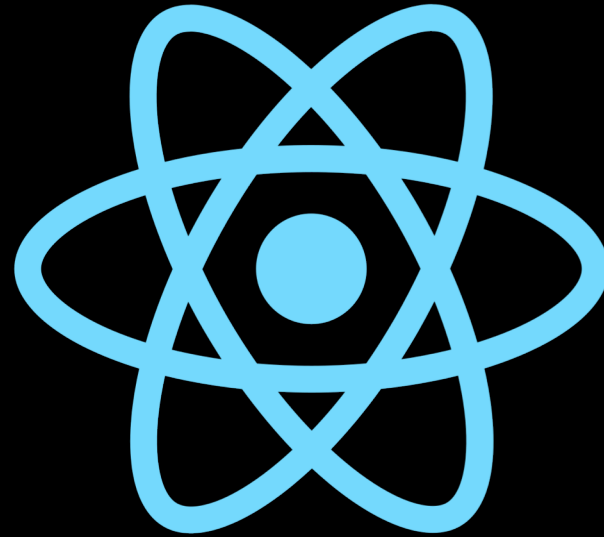
Do you really need to share state with your JavaScript app?

Maybe go with CSS modules instead...

Experience

React Project

- Video streaming platform
- ~100kLOC JS
- *styled-components*



styled-components

Pros

- Great and easy to style components
- Fast and efficient in development
- No conflicts
- Modular

Challenges

- SSR can be resource hungry
- Discipline required for high quality code
- Quickly pile up style duplication

Your experience?

Resources

Compare everything:

<https://michelebertoli.github.io/css-in-js/>

JSS

<https://cssinjs.org>

Styled-components

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

Emotion

<https://emotion.sh/>