

Your Next Week

Saturday June 6

9:00 AM

- **DUE Class 19 Mock Interviews**
- **DUE Class 19 Lab**
- *NO READING Class 20*
- Class 20A

Wednesday May 20

9:00 AM

- Class 20B

MIDNIGHT

- *NO LEARNING JOURNAL*
- Class 20*

Thursday May 21

6:30 PM

- Co-working

Friday May 22

Saturday May 23

9:00 AM

- Midterms Work
- **DUE Class 20 Lab**

Sunday May 24

MIDNIGHT

- **DUE Class 20-Midterms**
- Feedback**

Monday May 25

6:30 PM

- Optional Co-working

Tuesday May 26

6:30 PM

- Midterms work

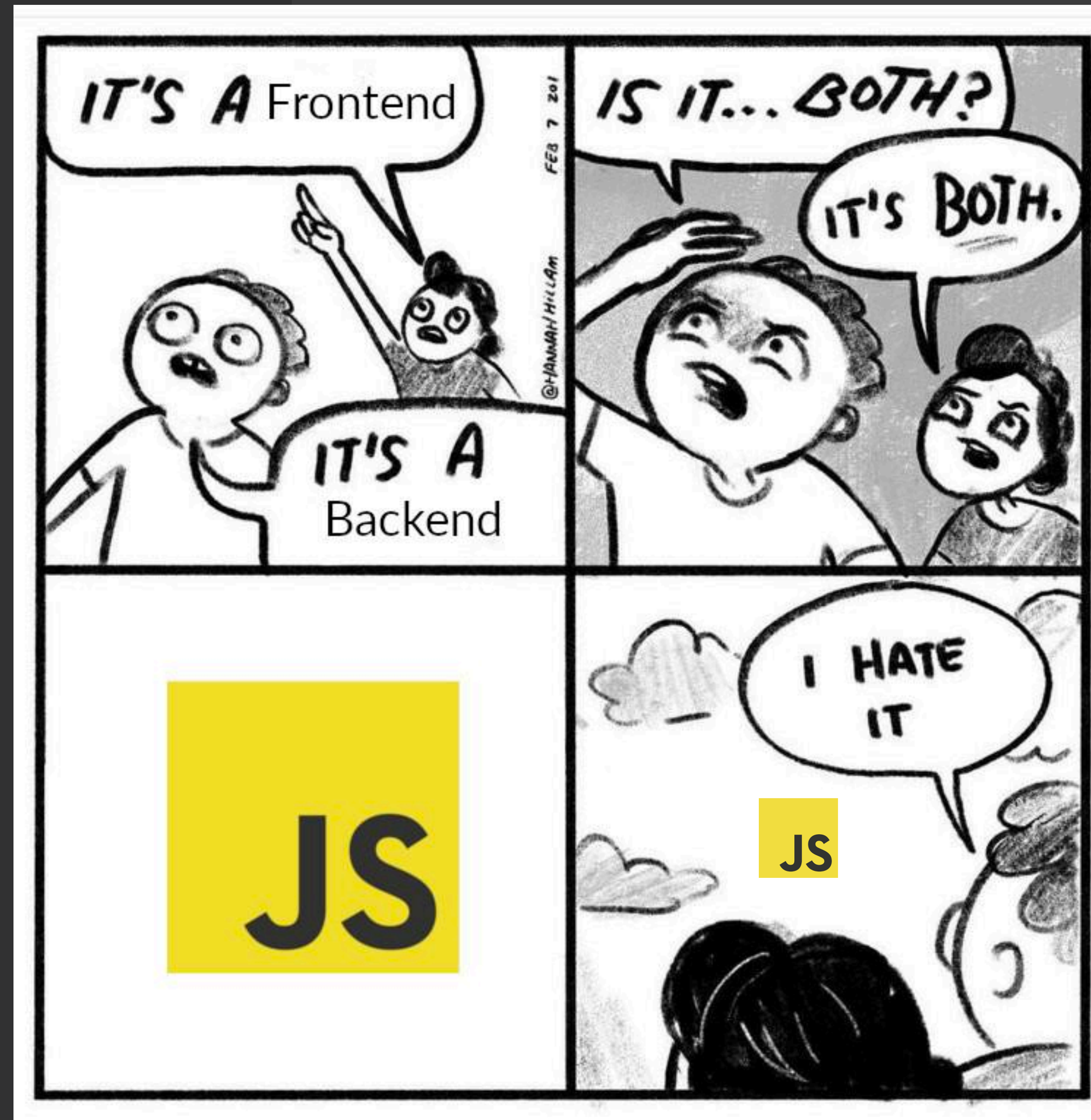
What We've Covered

<i>Module 01</i> Javascript Fundamentals and Data Models <i>C01 — Node Ecosystem, TDD, CI/CD</i> <i>C02 — Classes, Inheritance, Functional Programming</i> <i>C03 — Data Modeling & NoSQL Databases</i> <i>C04 — Advanced Mongo/Mongoose</i> <i>C05 — DSA: Linked Lists</i>	<i>Module 02</i> API Servers <i>C06 — HTTP and REST</i> <i>C07 — Express</i> <i>C08 — Express Routing & Connected API</i> <i>C09 — API Server</i> <i>C11 — DSA: Stacks and Queues</i>	<i>Module 03</i> Auth/Auth <i>C10 — Authentication</i> <i>C12 — OAuth</i> <i>C13 — Bearer Authorization</i> <i>C14 — Access Control (ACL)</i> <i>C15 — DSA: Trees</i>	<i>Module 04</i> Realtime <i>C16 — Event Driven Applications</i> <i>C17 — TCP Server</i> <i>C18 — Socket.io</i> <i>C19 — Message Queues</i> <i>C20 — Midterms Prep</i> <i>Midterms</i>
<i>Module 05</i> React Basics C21 — Component Based UI C22 — React Testing and Deployment C23 — Props and State C24 — Routing and Component Composition C25 — DSA: Sorting and HashTables	<i>Module 06</i> Advanced React C26 — Hooks API C27 — Custom Hooks C28 — Context API C29 — Application State with Redux C30 — DSA: Graphs	<i>Module 07</i> Redux State Management C31 — Combined Reducers C32 — Asynchronous Actions C33 — Additional Topics C34 — React Native C35 — DSA: Review	<i>Module 08</i> UI Frameworks C36 — Gatsby and Next C37 — JavaScript Frameworks C38 — Finals Prep Finals

You survived Midterms!!



Ready for Front End?



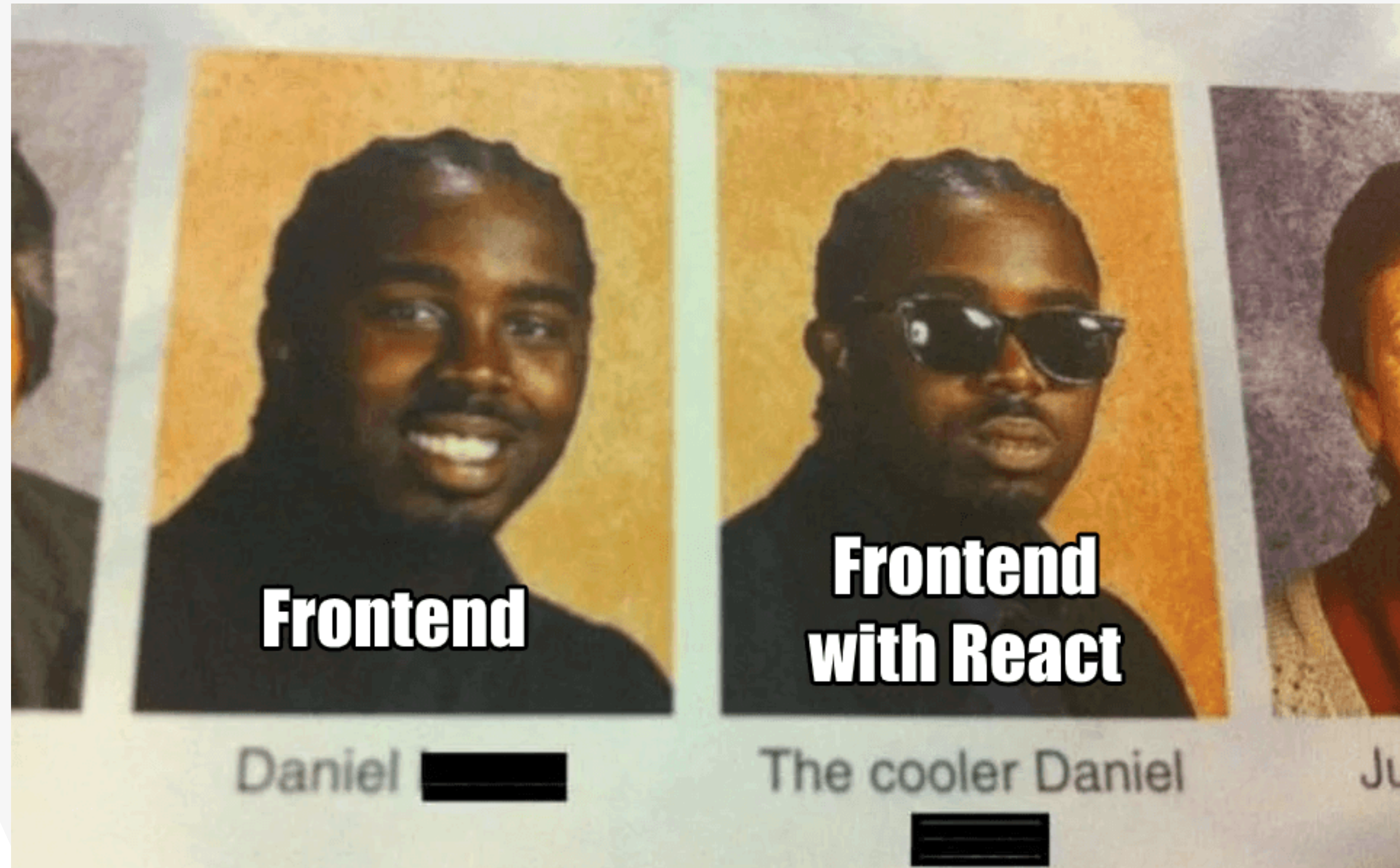
Class 21

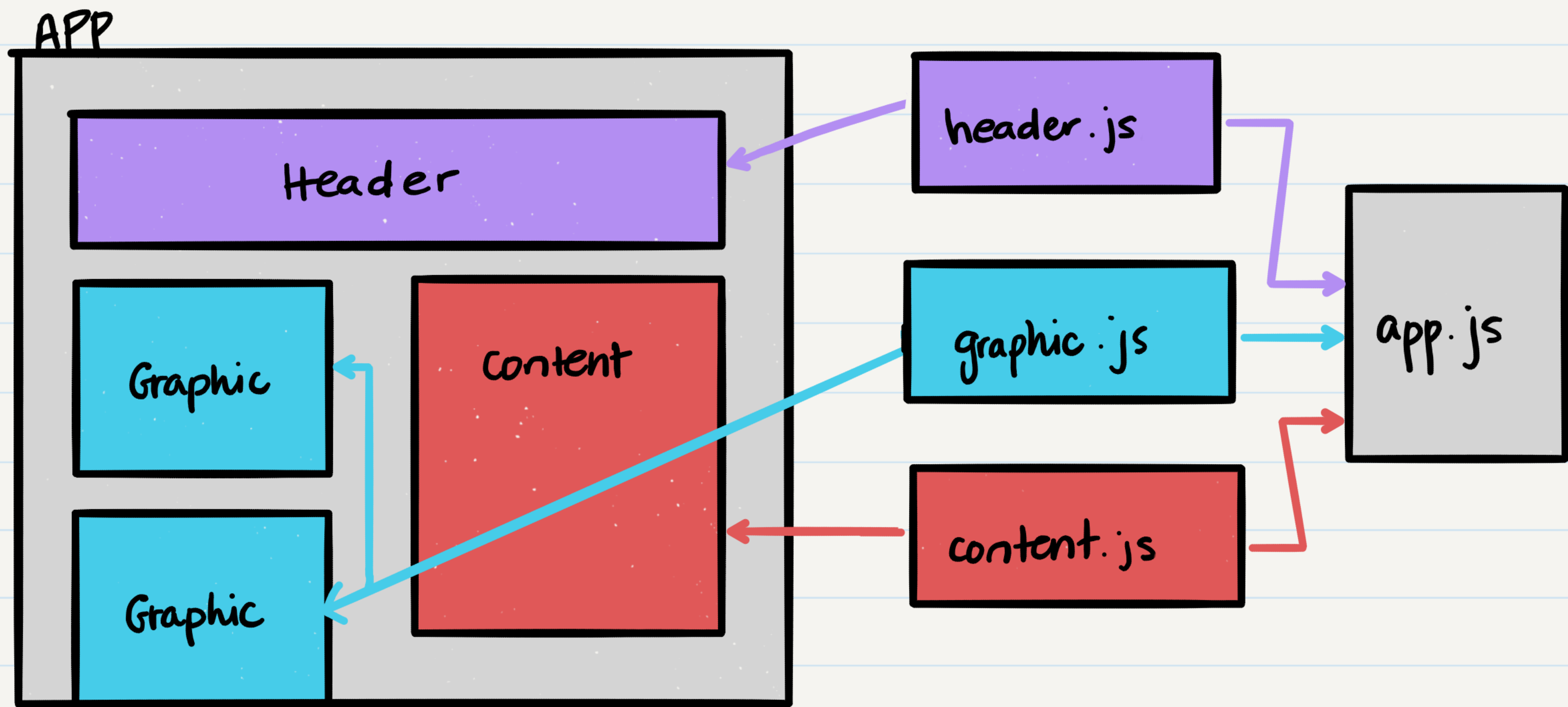
Component Based UI

seattle-javascript-401n16

Front End with React

- A JavaScript **library** by Facebook
- Combines JavaScript and HTML into one file (**JSX**)
- Let's us do cool JavaScript type stuff with our HTML
 - Modularize
 - Use variables
 - Dynamic updates





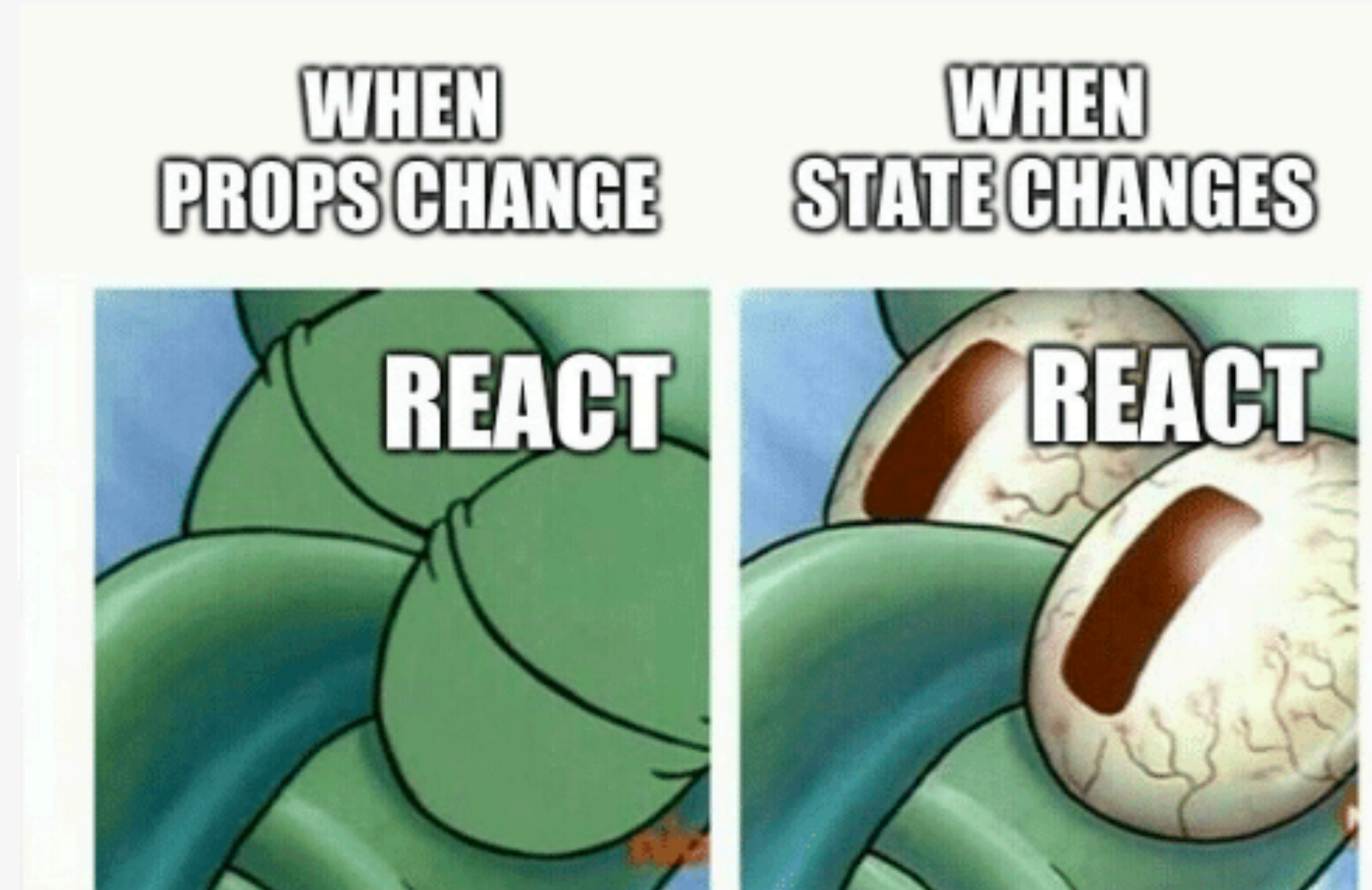
Components

- Pieces of UI that are self-contained/reusable
- They produce some dynamic HTML
- They can maintain their own **state**
 - A collection of local variables
 - When the state variables change, the HTML is re-generated / **re-rendered**
- A component is basically a dynamic HTML package! HTML + JavaScript logic



State vs Props

- State = **local variables** that cause the HTML to re-render on change
- Props = **parameter variables** that are given to this component from an outside source
 - Props don't effect render unless bound to state
- You can also have static (not stateful) local variables in a component

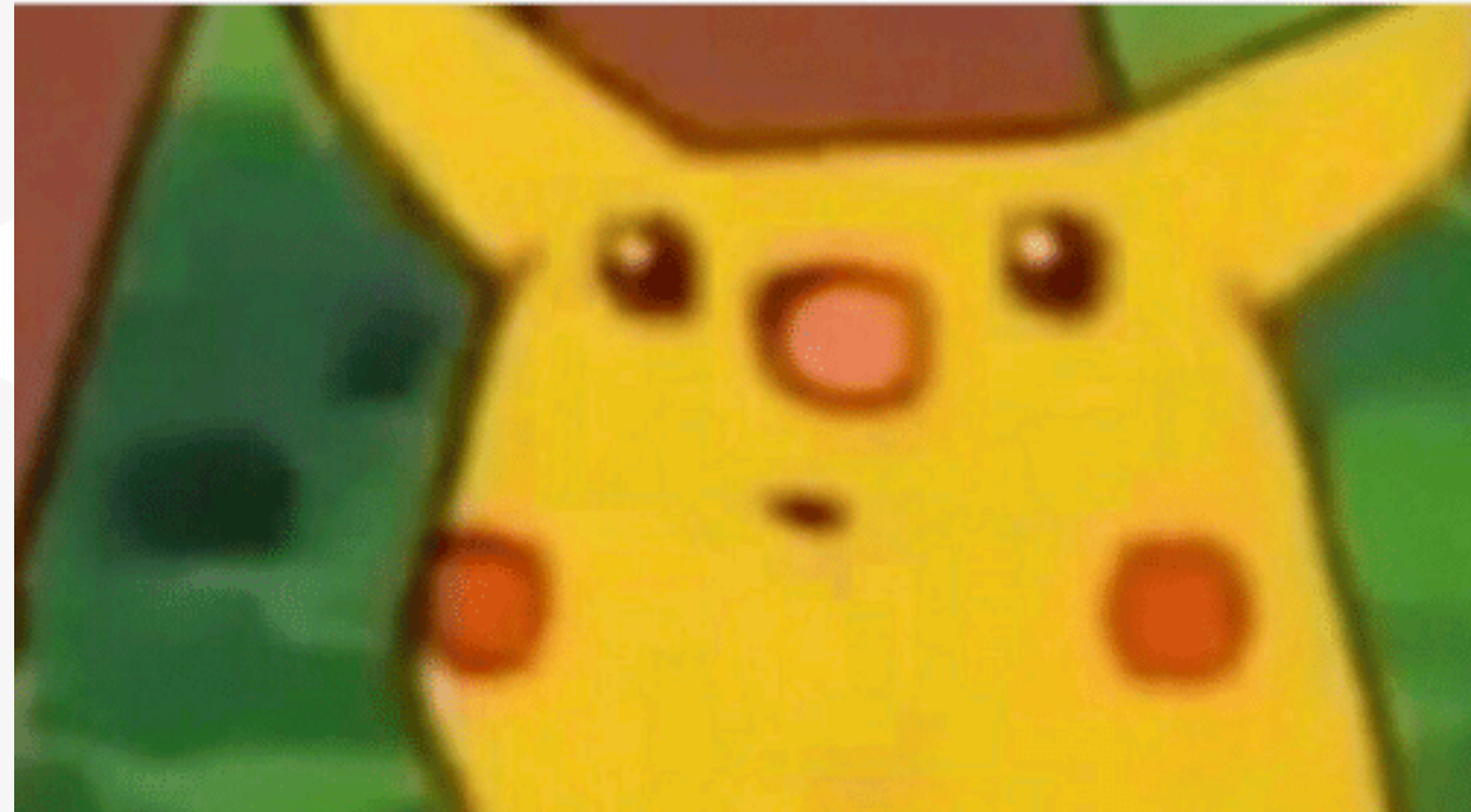


CSS Isn't Good

- Very restrictive
- Multiple sections of duplicated code
- Hard to change things
- Hard to separate pieces out / modularize
- No way to add code logic
- So...let's use a language that **compiles** into CSS!

Me: *makes small change to css*

My site:

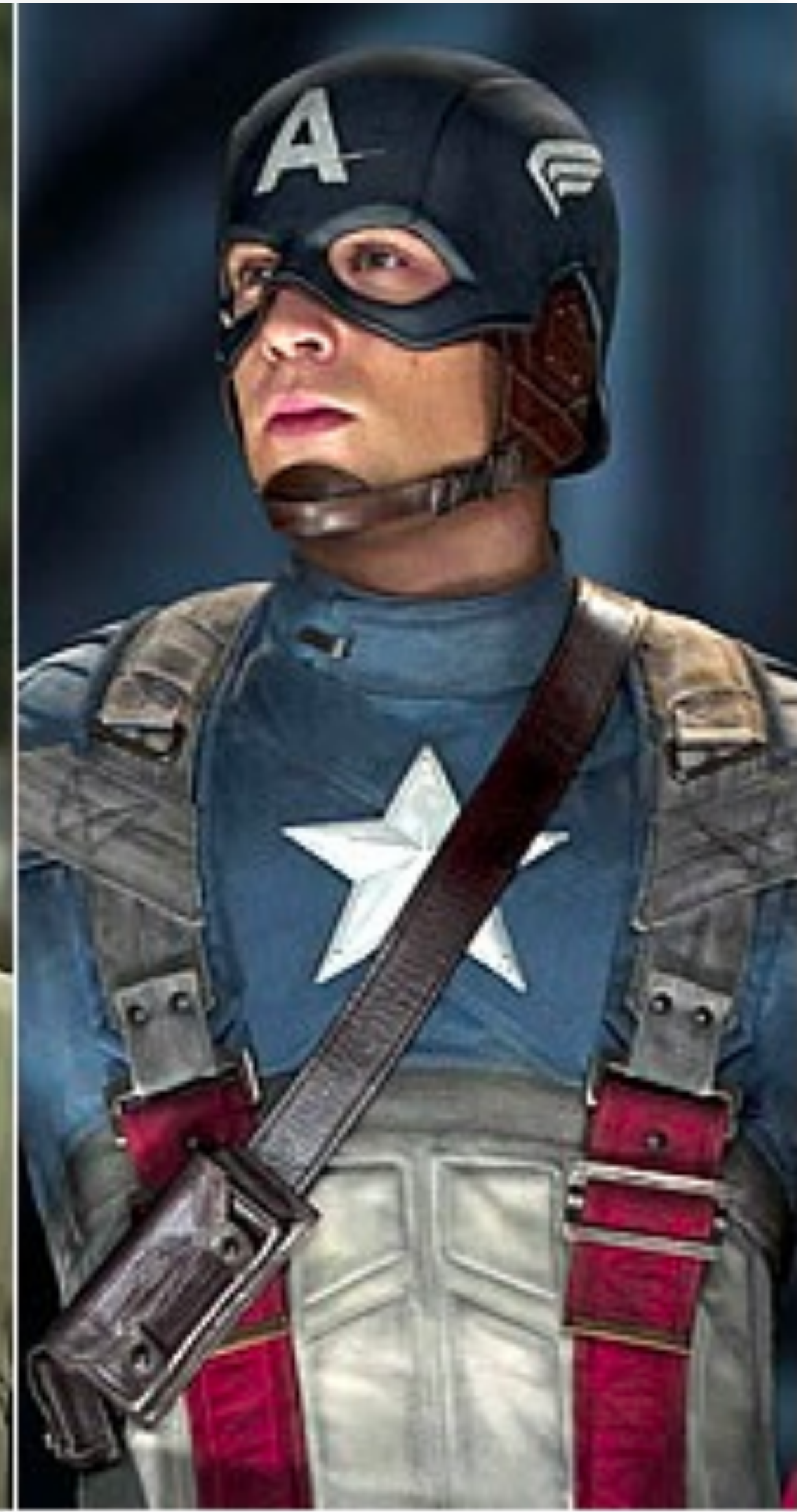


CSS Preprocessors (SASS, Less)

- Languages with more code-like features
- **Variables**! Change colors, spacing, fonts, etc across your whole UI effortlessly
- **Mixins**! Generate snippets of style / CSS which you can include anywhere
- **Functions**! Add, subtract, calculate style values
- **Partials**! Modularize your styles
- **Nesting**! Better organization of styles based on nested selectors



CSS



Sass

Lab 21 Overview