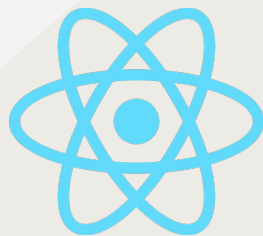


# React & Other JavaScript Libraries



CS 160 Summer 2022, Section 5A

by Lucy White

# Agenda

- Course reminders
- React Overview (15 min)
- JSX & Virtual DOM (10 min)
- React Props & States (10 min)
- React Resources (5 min)
- Explore JavaScript Libraries (10 mins)
- React Practice! (5 min)

# Week Overview

## Tuesday

4.4 Personas and  
Narratives due

## Wednesday

4.5 Dialogue  
Flows due

## Thursday

4.6 Experiment  
Design Due

## Friday

4.7 Figma  
Prototype due

Quiz 5 due

4.8a Participate in  
Usability Testing

# Course Reminders

## Grades Released

- All grades for project 1 and 2 have been released
- [Regrade Request Form](#)

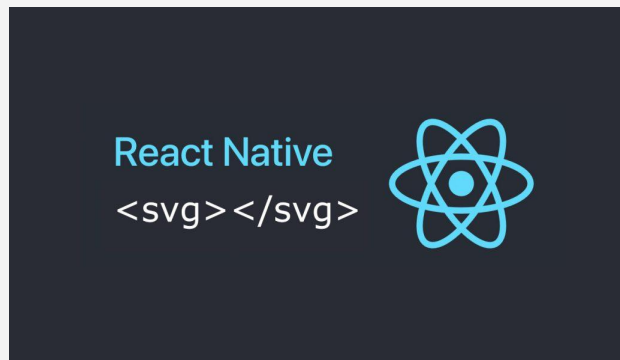
## Project 4 Partner Problems

- Fill out this [form](#) if you're having communication issues with your assigned partner

## Special Lecture!

- Wed 7/20: guest lecture from Dr. Kishonna L. Gray

# What is React?

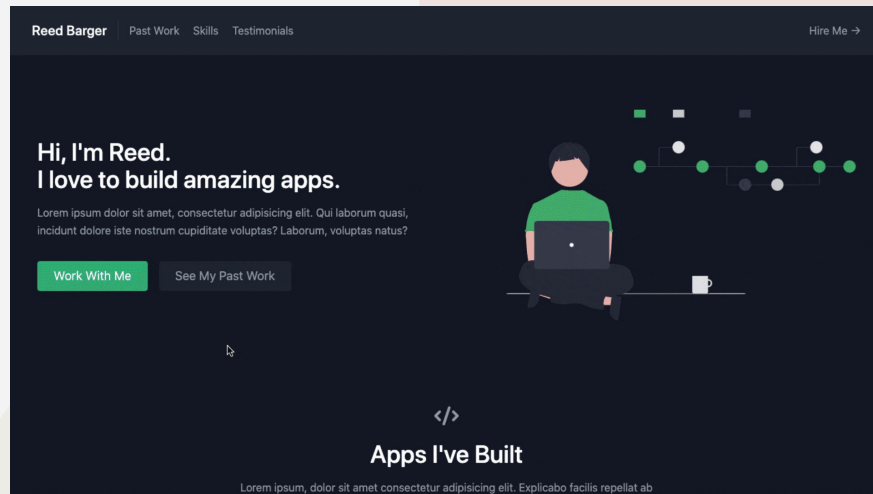
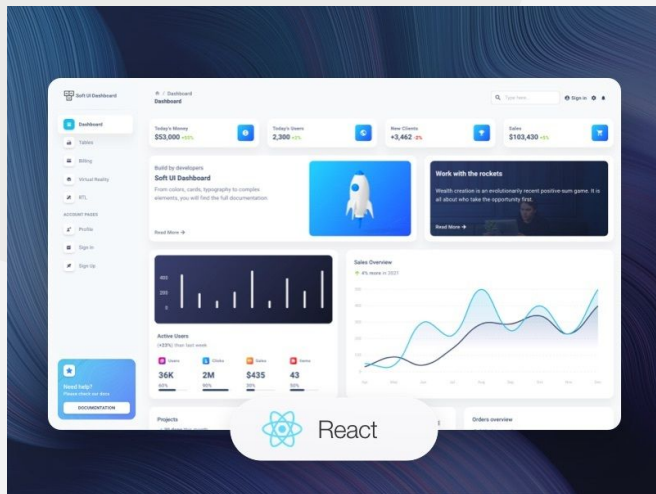


- Created in 2013 by a Facebook engineer
- JS library used in web development to build interactive elements on websites
- Used to develop both web and mobile applications! (React Native)

# Why React?

- Easier to create dynamic web applications (more functionality + less coding!)
- Improved performance: uses virtual DOM, only rerenders and updates new changes in a web page
- Reusable Components: every component has its own logic and controls and reusing them reduces development time
- Unidirectional data flow - data flows in a **single** direction, easier to debug!

# React Examples!



# JSX Example

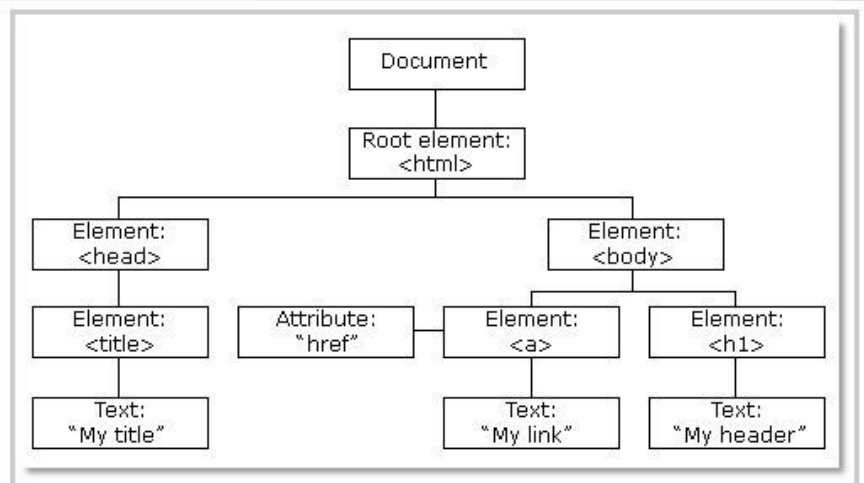
- JSX is a JavaScript syntax extension that is used with React to describe what the UI should look like
- React doesn't require JSX, but people find it convenient to use with React and it allows React to show helpful error and warning messages

```
const name = 'Josh Perez';  
const element = <h1>Hello, {name}</h1>;
```

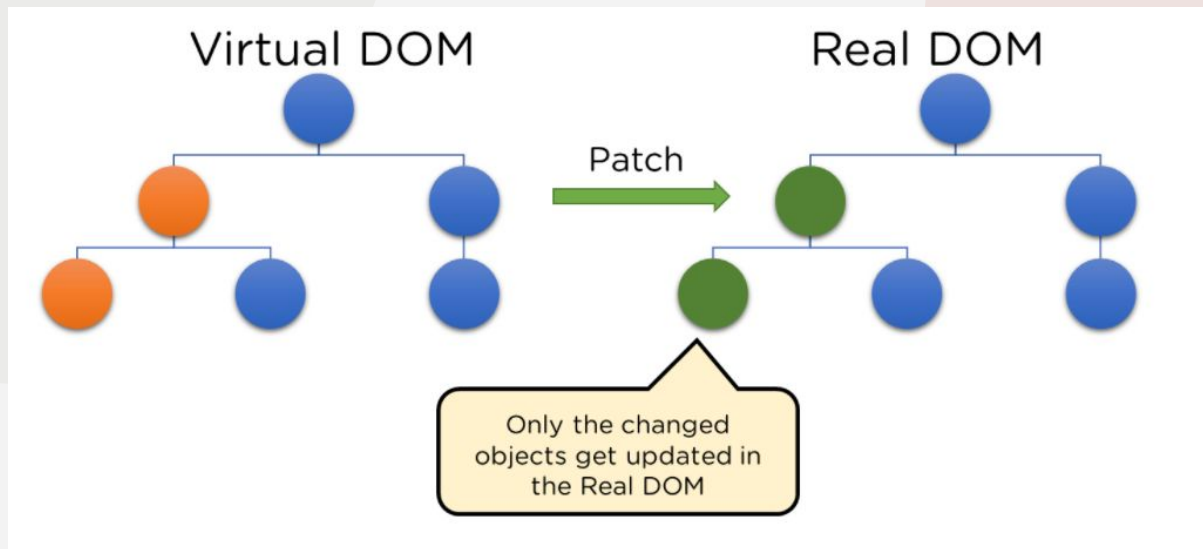


# Document Object Model (DOM)

a.k.a Real DOM

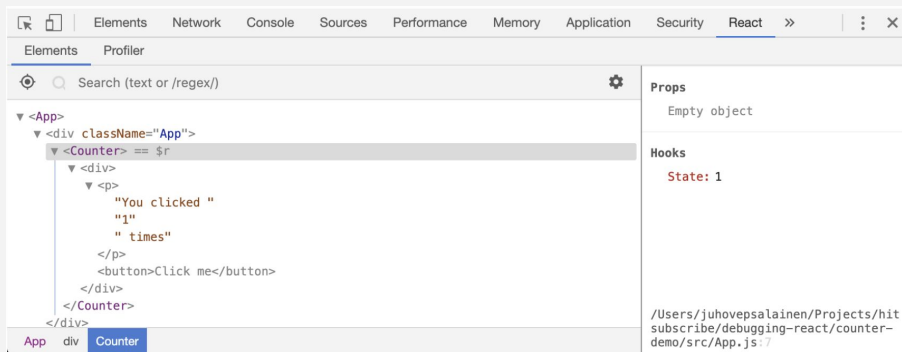


# Virtual DOM



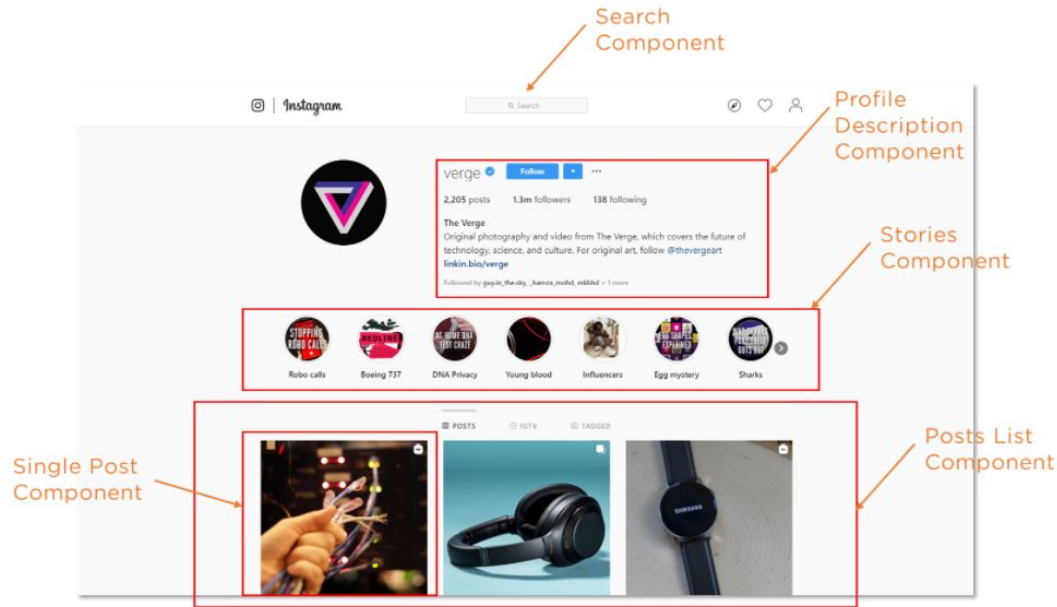
- React DOM takes care of updating the DOM to match the React elements
- To render a React element, first pass the DOM element to [ReactDOM.createRoot\(\)](#), then pass the React element to `root.render()`
  - a. [Render example](#)
  - b. [Update example](#)

# Debugging React



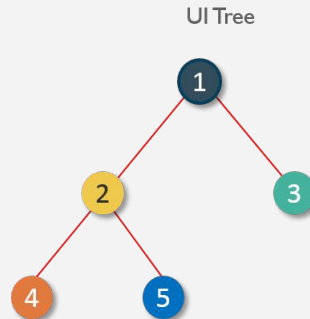
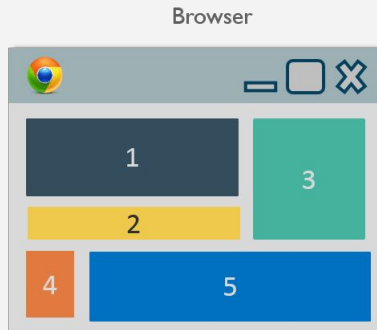
- Tool: React Developer Tools
  - An extension for Chrome-based browsers and Firefox that implements React-specific debugging functionality
  - React debugging guide

# React Components

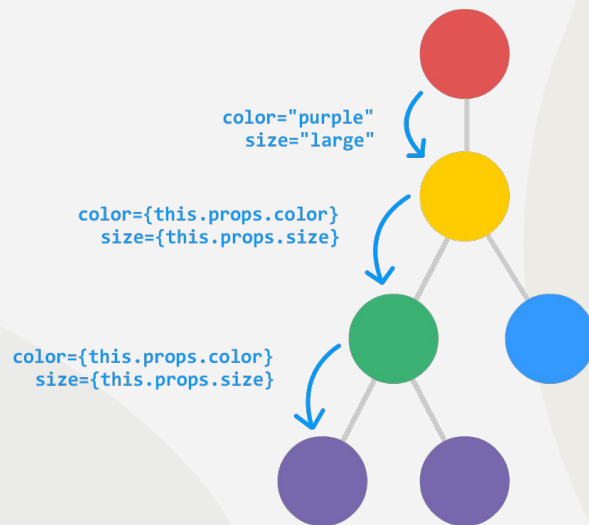
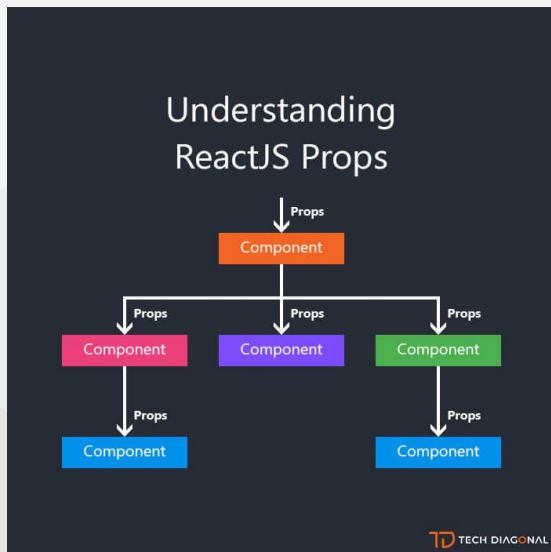


# Features of Components

- **Reusability** - components can be reused in multiple areas and files
- **Nesting** - A component can contain multiple child components
- **Input** - accept arbitrary “props” (properties) as input
- **Output** - return React elements describing what should appear on the screen



# React Props



# Component Example

- Always start components with a capital letter!
- Syntax for invoking a component: `<Component_name props />`
- Example:
  - We call `root.render()` with the `<Welcome name="Sara" />` element.
  - React calls the `Welcome` component with `{name: 'Sara'}` as the props.
  - Our `Welcome` component returns a `<h1>Hello, Sara</h1>` element as the result.
  - React DOM efficiently updates the DOM to match `<h1>Hello, Sara</h1>`.

# React States

```
1 class Clock extends React.Component {  
2   constructor(props) {  
3     super(props);  
4     this.state = {date: new Date()};  
5   }
```

[“tick” example](#)

[Stateless clock](#)

[State clock](#)

- Like props, but fully controlled by the component
- Object that stores properties values that can change
- Re-renders component to browser
- Multiple properties can be stored in state object



# Props vs States



## Props vs State



✓ props are read-only

✓ props can not be modified

✓ state changes can be asynchronous

✓ state can be modified using `this.setState`

# React JS Resources!

## Example Projects

There are many example projects created by the React community. Feel free to add your own project. If you add a project, please commit to keeping it up to date with the latest versions of React.

- **Calculator** Implementation of the iOS calculator built in React
- **Emoji Search** Simple React app for searching emoji
- **Github Battle App** Battle two Github users and see the most popular Github projects for any language.
- **React Powered Hacker News Client** A React & react-router-powered implementation of Hacker News using its Firebase API.
- **Pokedex** The list of Pokémon with live search
- **Shopping Cart** Simple ecommerce cart application built using React
- **Progressive Web Tetris** Besides a beautiful, mobile-friendly implementation of Tetris, this project is a playground for integrating and experimenting with web technologies.
- **Product Comparison Page** Simple Product Compare page built in React
- **Hacker News Clone React/GraphQL** Hacker News clone rewritten with universal JavaScript, using React and GraphQL.

# Extra JavaScript Libraries



Angular JS



D3

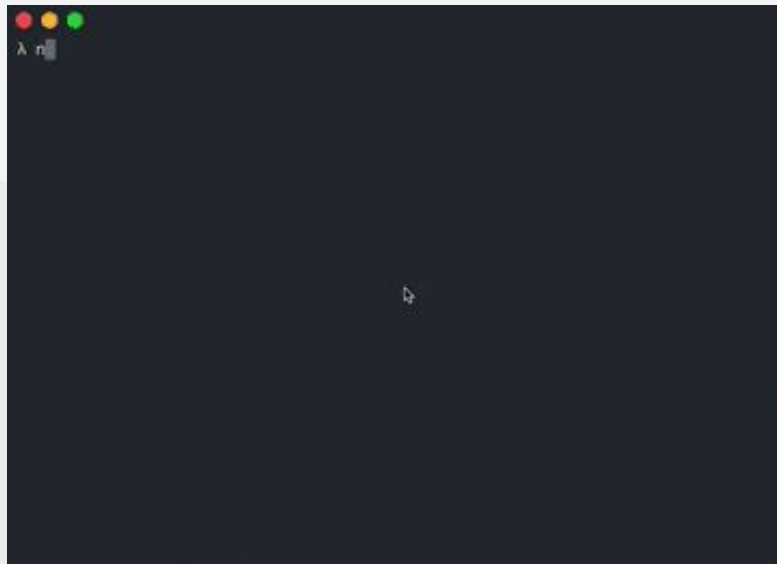


ChartJS

# React Practice!

## Create Your Own React App

alternatively : look  
into other JS libraries  
you might like to use  
for your projects



[Hello World! starter code](#)