

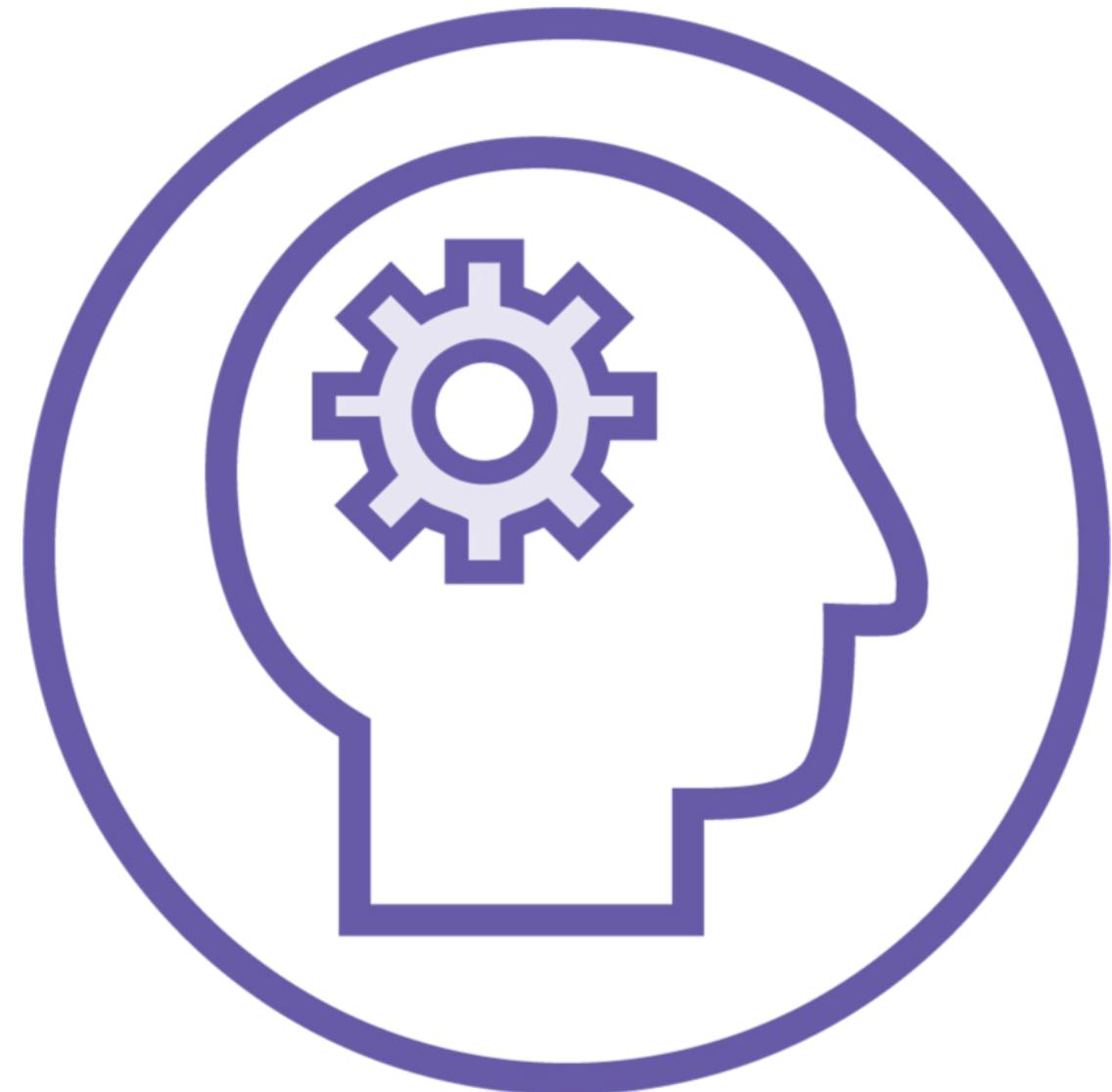
Getting Started to Build Applications with React



Adhithi Ravichandran
Software Consultant, Author, Speaker

@AdhithiRavi www.adhithiravichandran.com





Prerequisites

- **JavaScript**
- **ES2017 basic knowledge**
- **HTML**
- **CSS**



Code in JavaScript



According to Stack Overflow's Annual Survey of 2022, for the 10th year in a row, JavaScript is ranked as the most commonly used programming language.



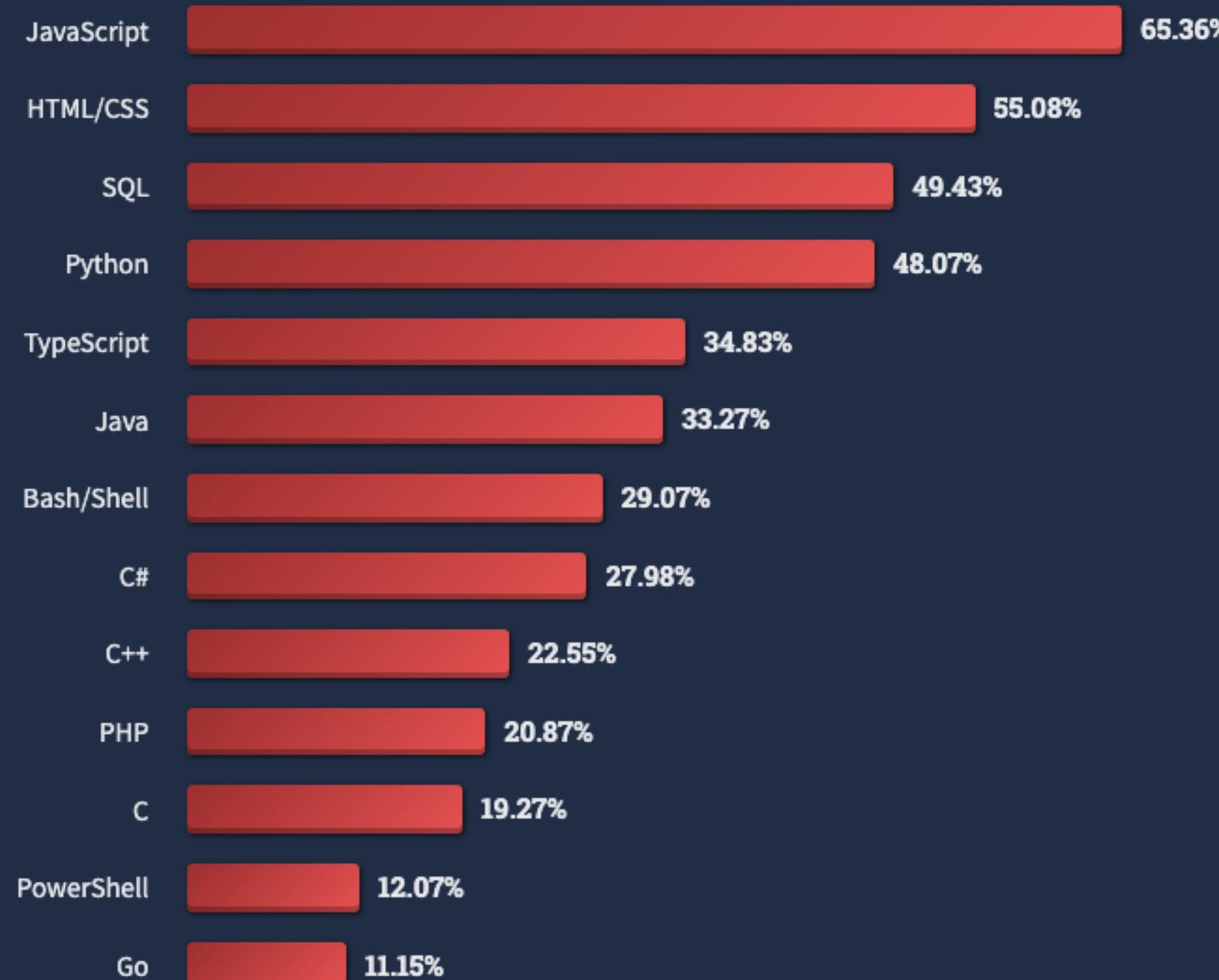
Stack Overflow Survey Results 2022

All Respondents

Professional Developers

Learning to Code

71,547 responses



What is React?



React is a JavaScript library for
building rich user interfaces.





Most Wanted Web Frameworks 2022

React.js completes its fifth year as most wanted in the stack overflow developer survey!

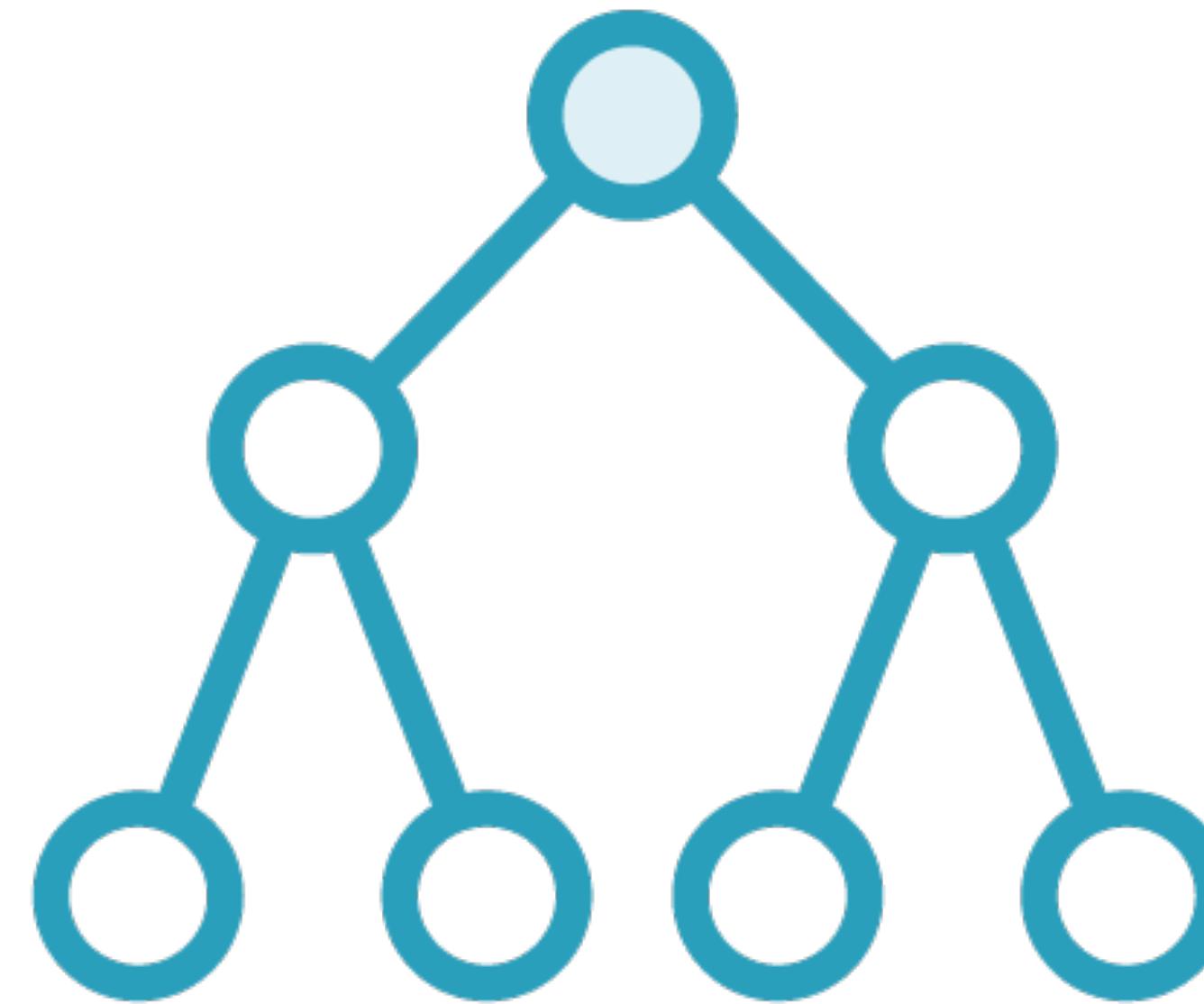


**Component-based
architecture**

Reuse of components

One directional data flow

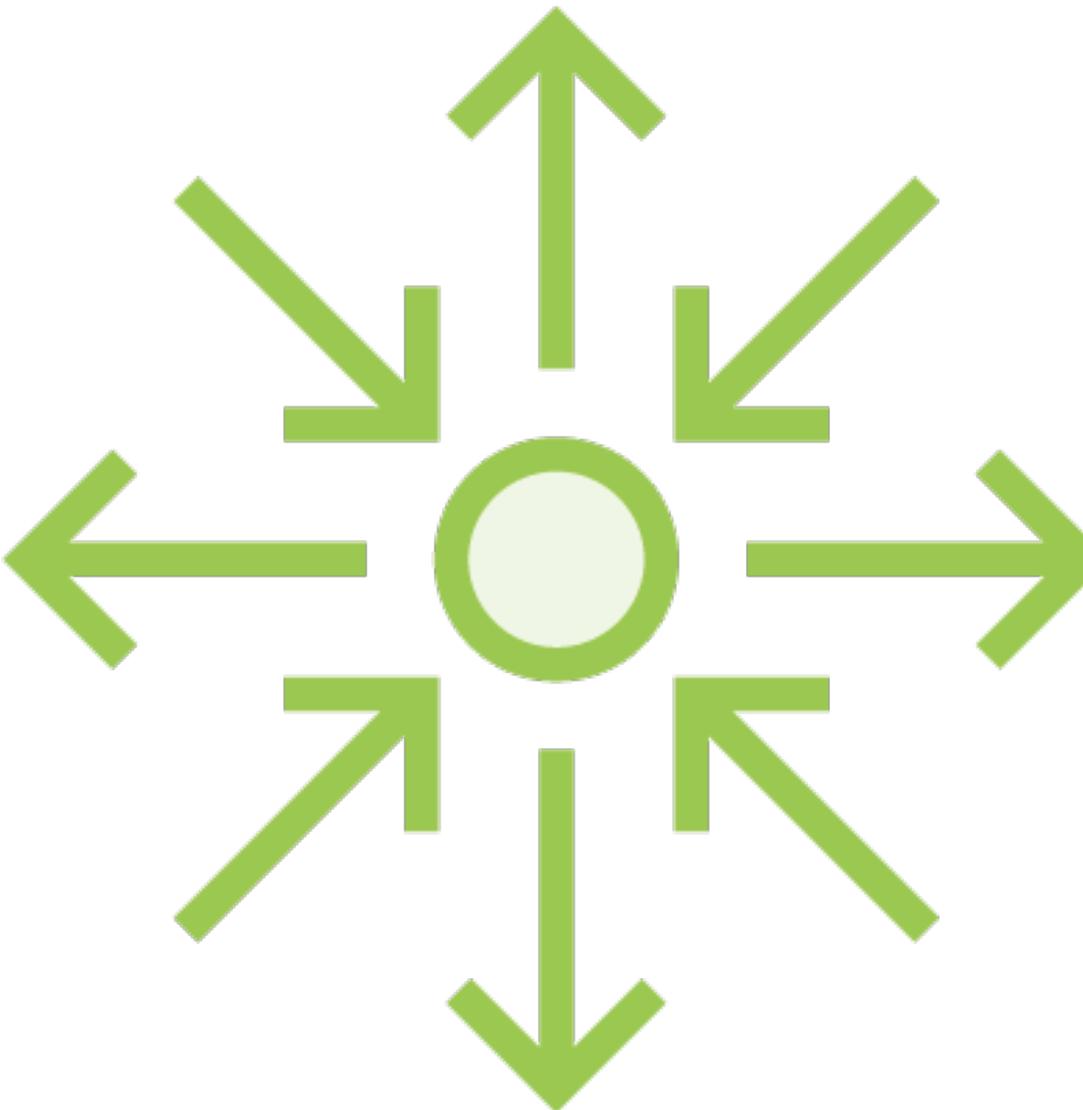
Everything Is a Component



Offers lots of flexibility

**Freedom to choose
state management,
navigation, server-side
rendering and more**

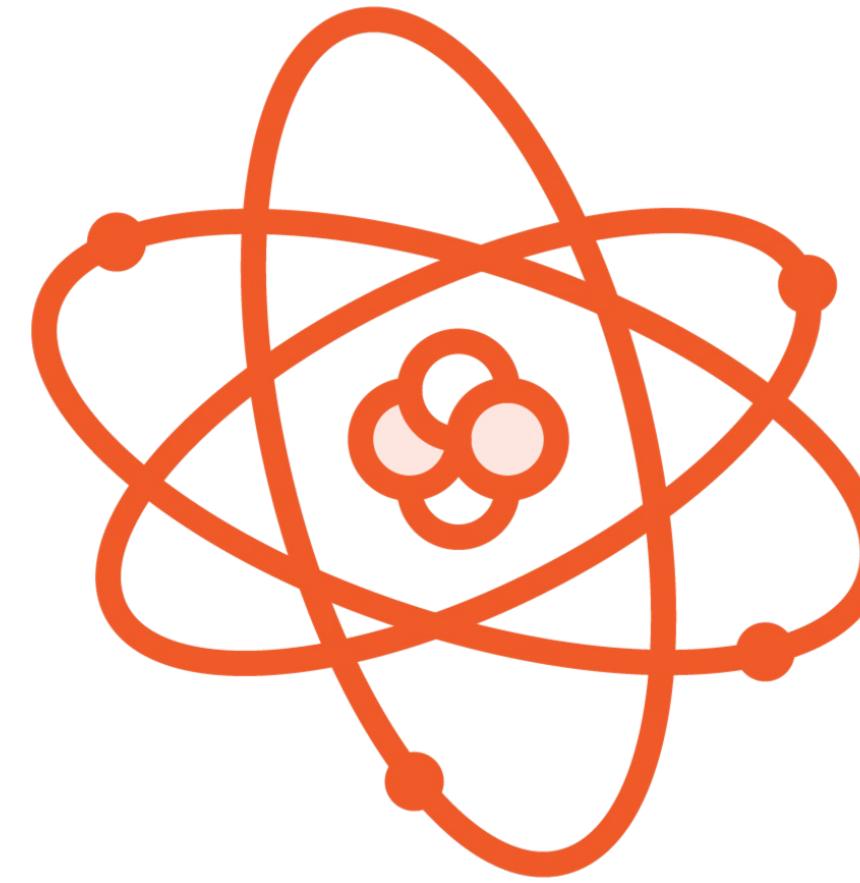
Flexibility



Framework vs. Library



Angular



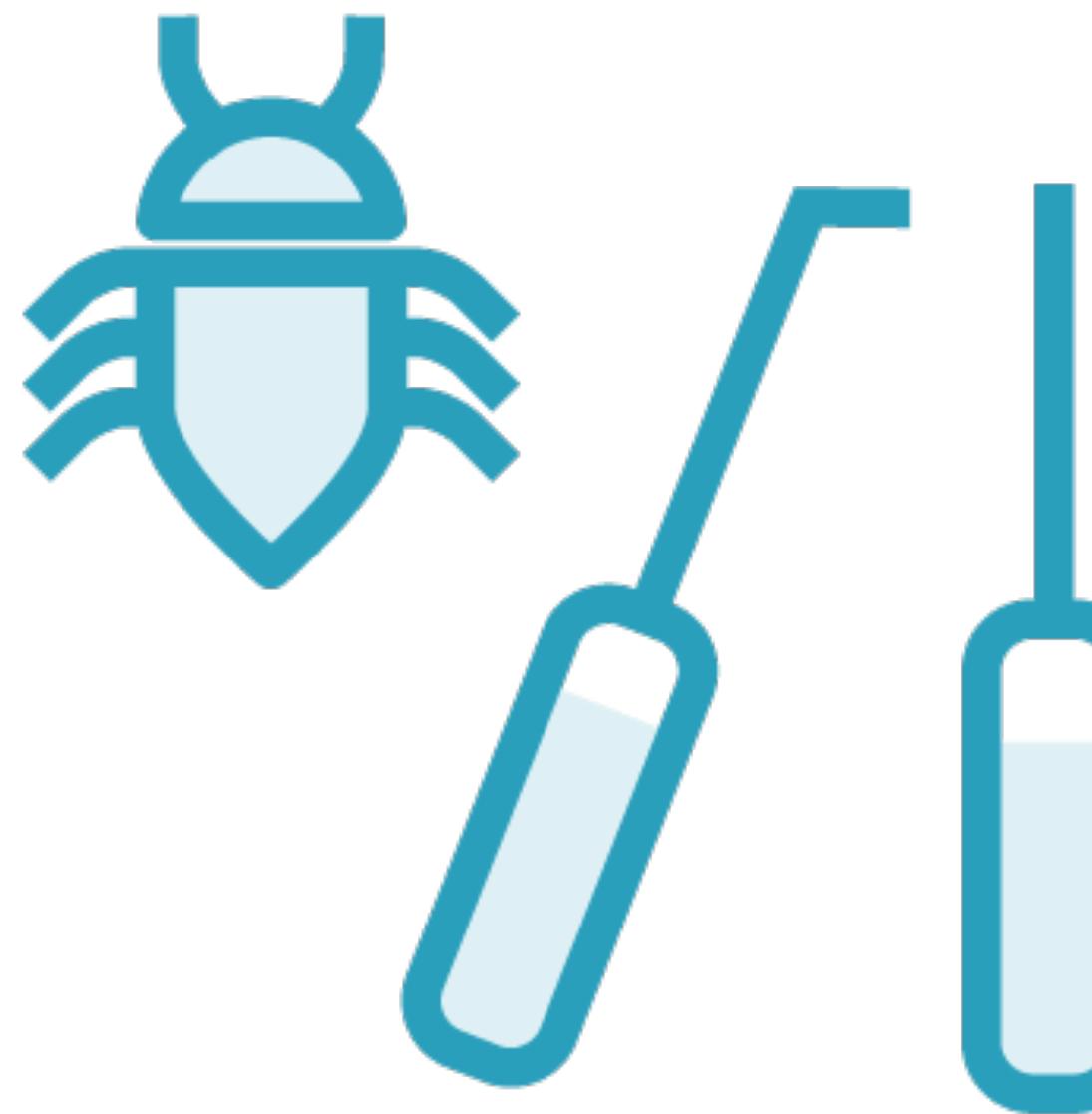
React



Testing Is Fun

Ease of testing

**Plenty of testing
options like Jest,
Cypress, etc..**



Builds Rich UI



Seamless UI

**Highly responsive and
rich UI**

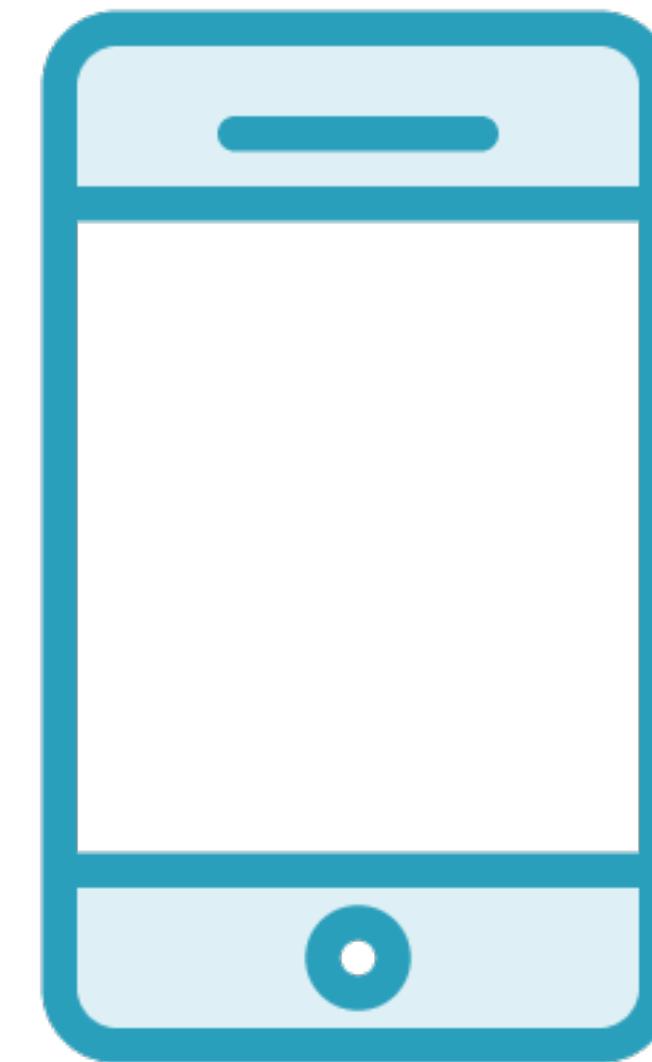
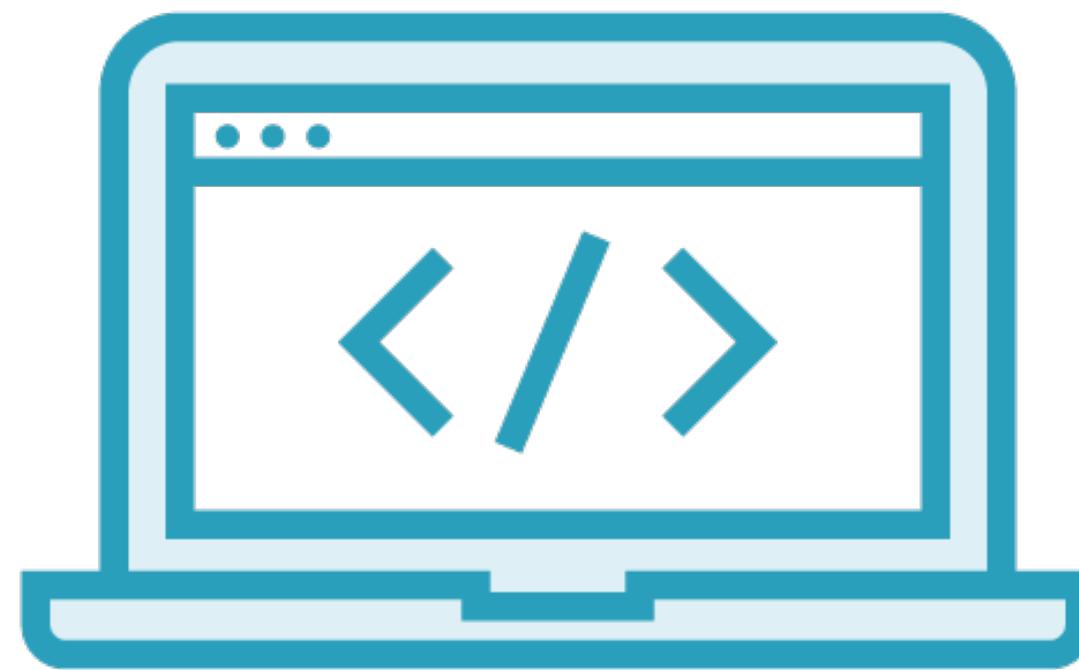
**Quick loading times
enabling a smooth
user experience**



Code Sharing

**Code sharing between
web and mobile**

**Saves cost of
development and
maintenance**



React Components



Components

Components let you split the UI into independent, reusable pieces that are easier to write, test, and maintain.





Search Twitter

[Home](#)[Explore](#)[Notifications](#)[Messages](#)[Bookmarks](#)[Lists](#)[Profile](#)[More](#)[Tweet](#)

you

#Tokyo2020

Trending

COVID-19

News

Sports

Entertainment



Bloomberg Quicktake · August 13, 2021

Fencing gains popularity in Hong Kong after Olympic gold win

Bloomberg Quicktake · August 12, 2021

This Olympian won gold after a #Tokyo2020 🏟 volunteer paid for his taxi when he went to the wrong venue



Sports Insider · August 12, 2021

Meet the man in charge of timing the Olympics



Bloomberg Quicktake · August 9, 2021

Tokyo 2020 Olympians receive a hero's welcome from hometown fans



Who to follow



Mary Grygleski
@mgrygles

[Follow](#)

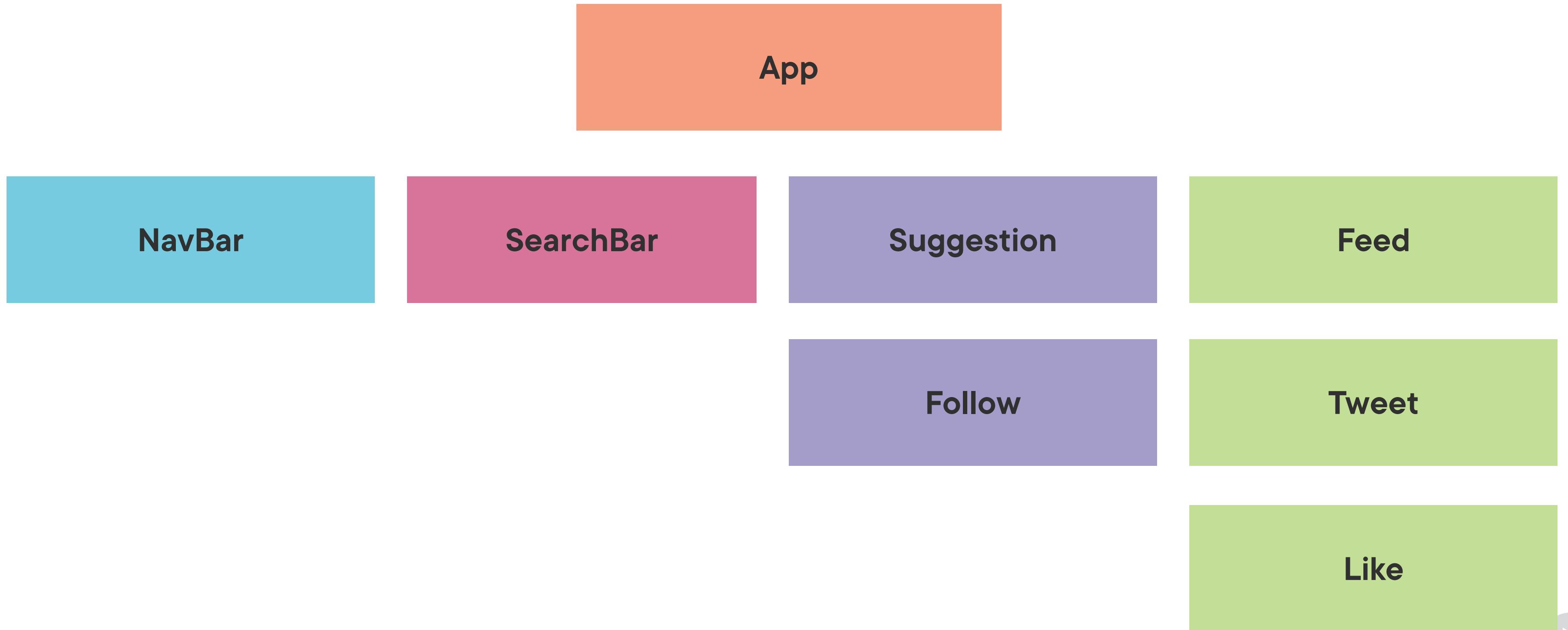
Developer Relations
@JobsInDevRel

[Follow](#)

Kate Inyeong Kim
@kateinkim

[Follow](#)[Show more](#)[Terms of Service](#) [Privacy Policy](#) [Cookie Policy](#)[Ads info](#) [More ...](#) © 2021 Twitter, Inc.

Components Structure



Each component is a building block that is a reusable piece of UI. Putting them together results in a complete application!



```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello Welcome to this presentation on React</h1>;  
  }  
}
```

React Component

Define a component as an ES6 class

```
function Welcome() {  
  return <h1>Hello Welcome to this presentation on React</h1>;  
}
```

React Component

Define a component with a JS function.

```
function formatName(user) {  
  return user.firstName + ' ' + user.lastName;  
}  
  
const user = {  
  firstName: 'Harper',  
  lastName: 'Perez',  
};  
  
const element = <h1>Hello, {formatName(user)}!</h1>;  
  
ReactDOM.render(element, document.getElementById('root'));
```

JSX

JSX - JavaScript Expressions, not HTML

It is a syntax extension to JavaScript

Can use JSX inside if statements, for loops, accept is as arguments, and return it from functions!

Use Functional Components





Class components are a story of the past

**Class components are confusing to people
and machines**



Write React Functional Components



Simpler to learn and write

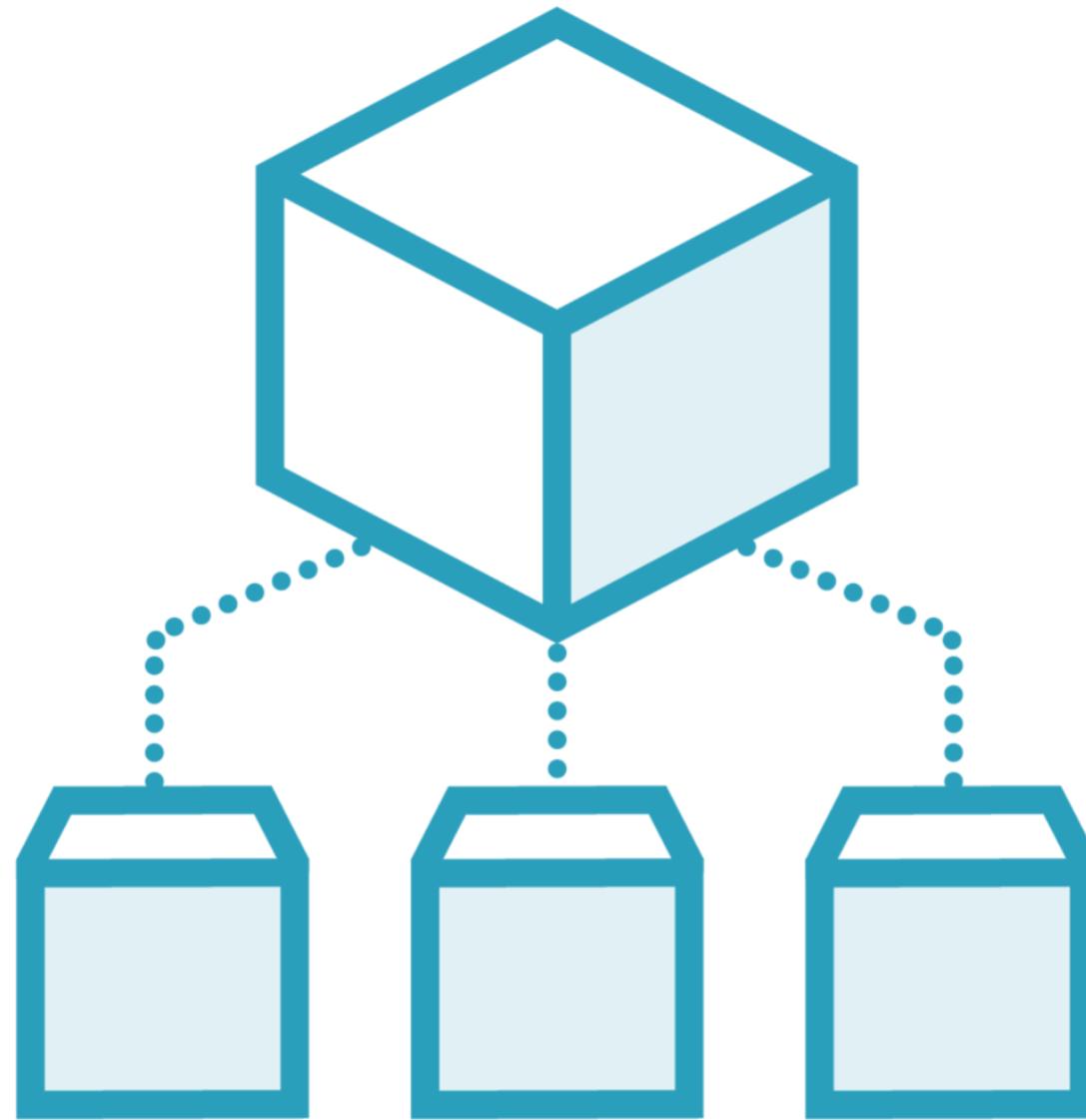
Less code

Easier to maintain and test

Hooks



Props



Props are short for *Properties*.

Components receive *props* from their parent.

Props are unchangeable within the component using it.

It helps you write reusable code.



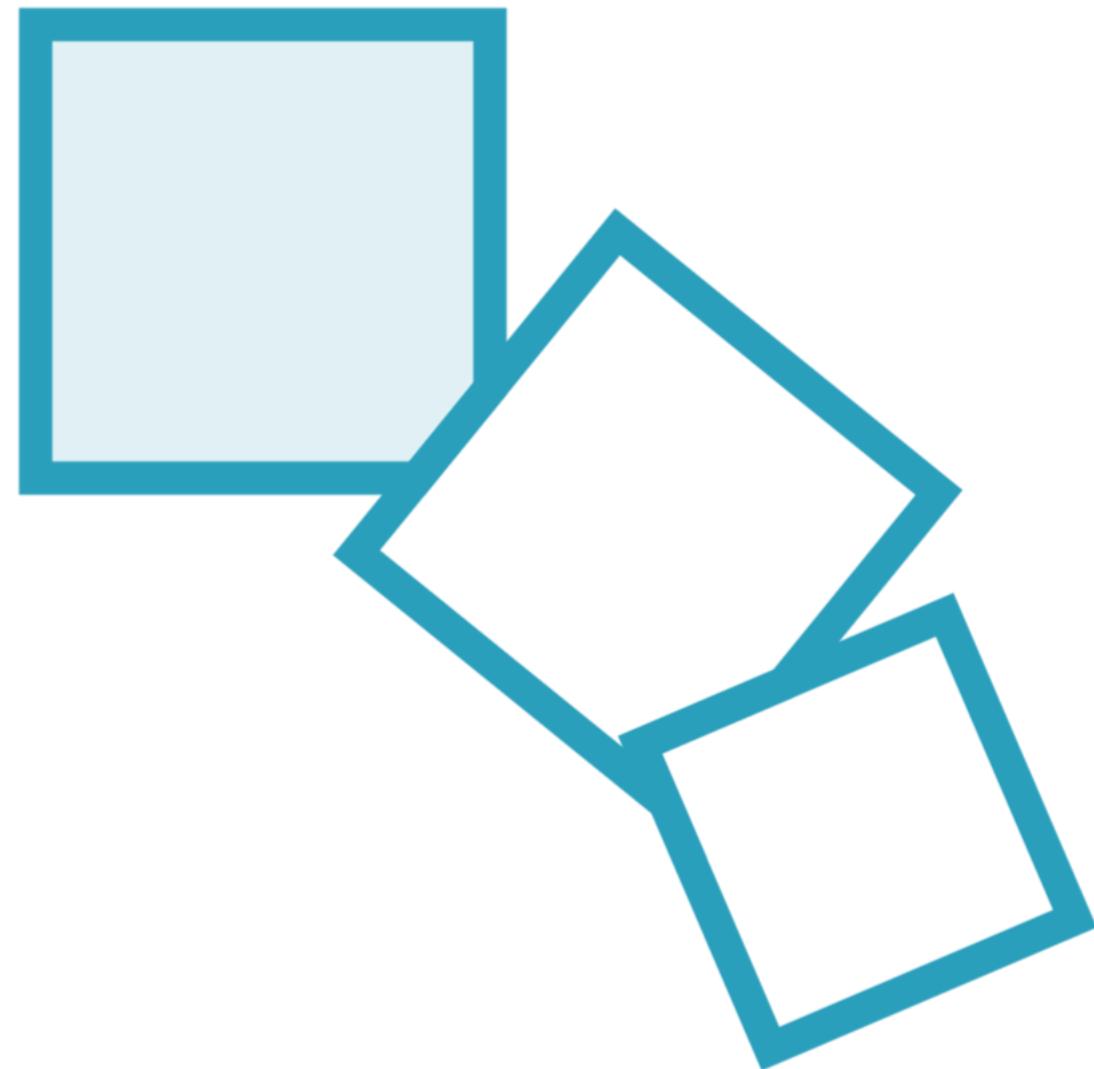
```
function Welcome(props) {  
  return <h1>Hello, {props.name} Welcome to this presentation on React</h1>;  
}
```

```
function App() {  
  return (  
    <div>  
      <Welcome name="Sara" />  
      <Welcome name="Adhithi" />  
      <Welcome name="Joe" />  
    </div>  
  );  
}
```

Passing Props to Components

- **Welcome** component accepts *props*.
- **App** component calls **Welcome** and passes *props*.
- Different *props* can be passed each time **Welcome** is called.
- *Props* are ready-only inside **Welcome** component.

State



State is internal to a component.

State can change within the component.

**Anytime there is data that is going to change
within a component – use state!**



```
import React, { useState } from 'react';

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Using State within Components

State is internal to component and can change within the component.

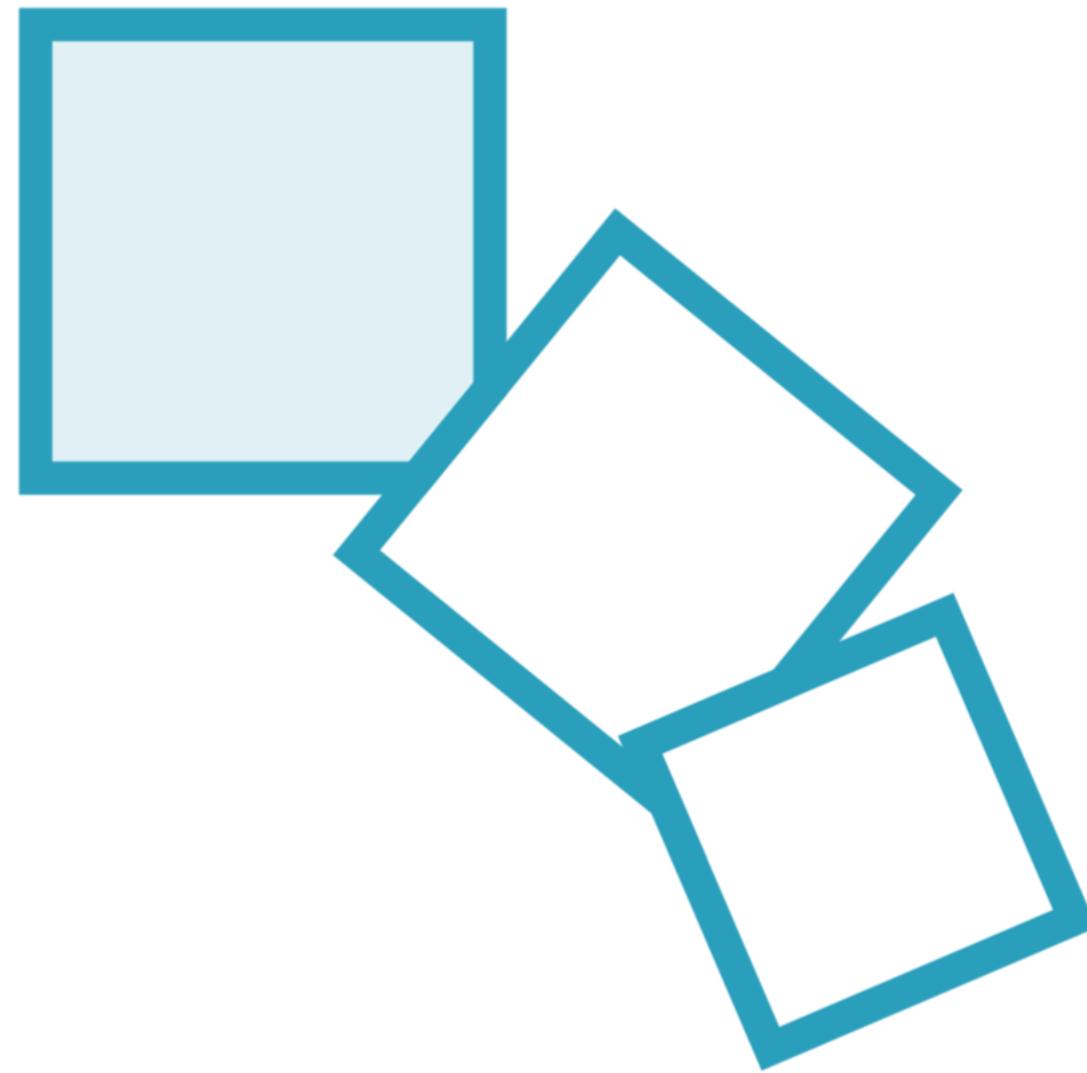
React – “reacts” to state
changes!



Hook is a special function that let's you hook into React features, within functional components!



useState Hook



Declares state variable.

Returns pair of values: current state and the function that updates it.



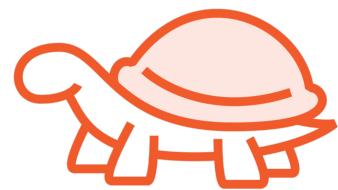
**Break down your
components – when needed!**



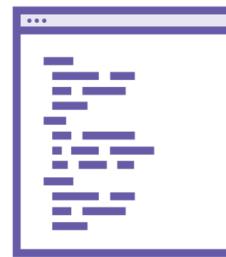
When do I break down components?



Managing state is a nightmare



Performance concerns with re-rendering of application



Code readability takes a hit



Working with multiple developers on the codebase becomes challenging



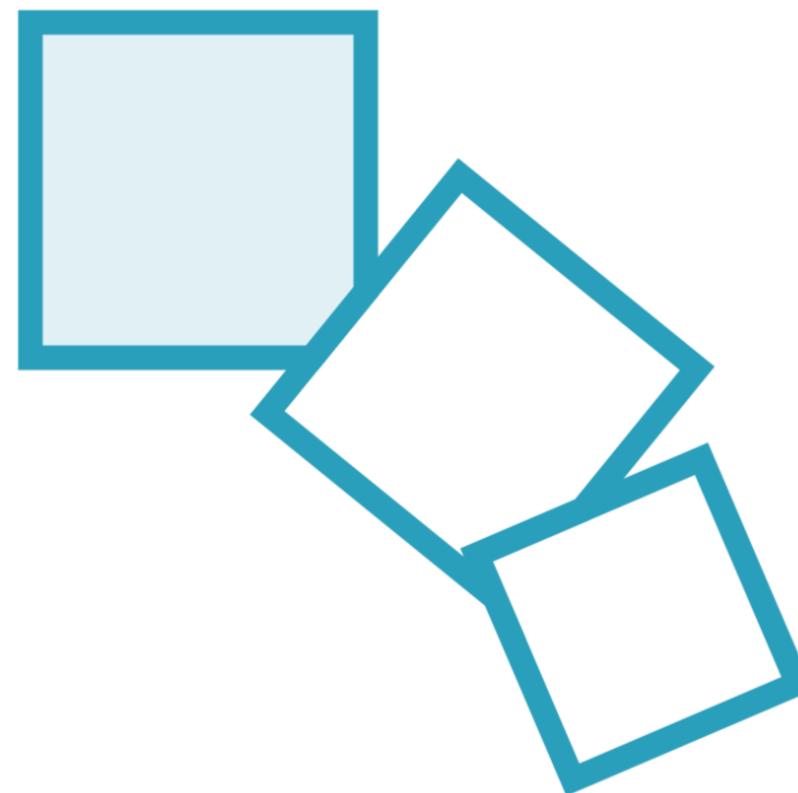
Testing code is harder



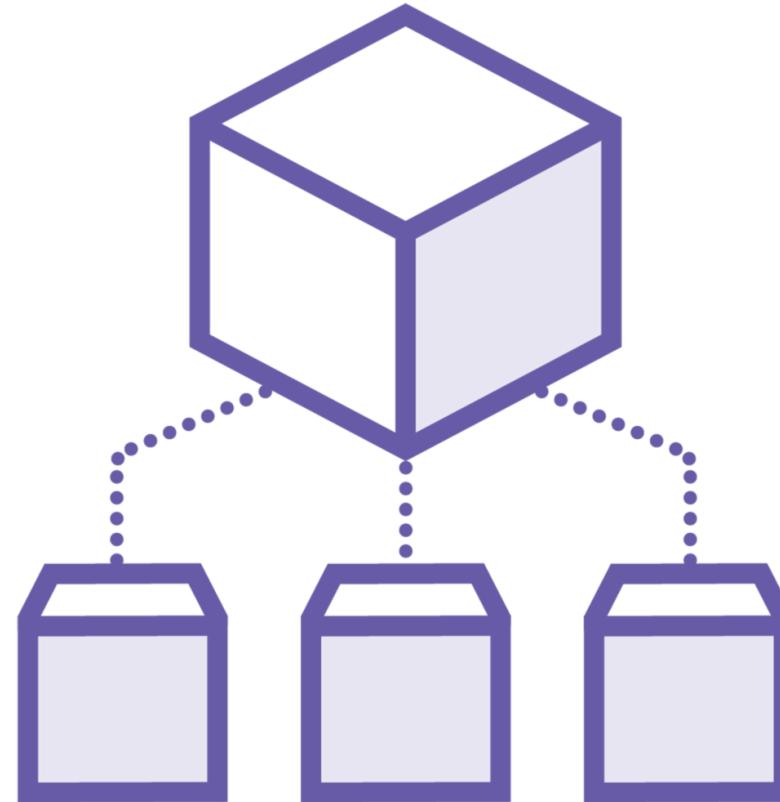
Start with local state!



State Management



Start with local state first



Pass down state via *props* if child component needs it



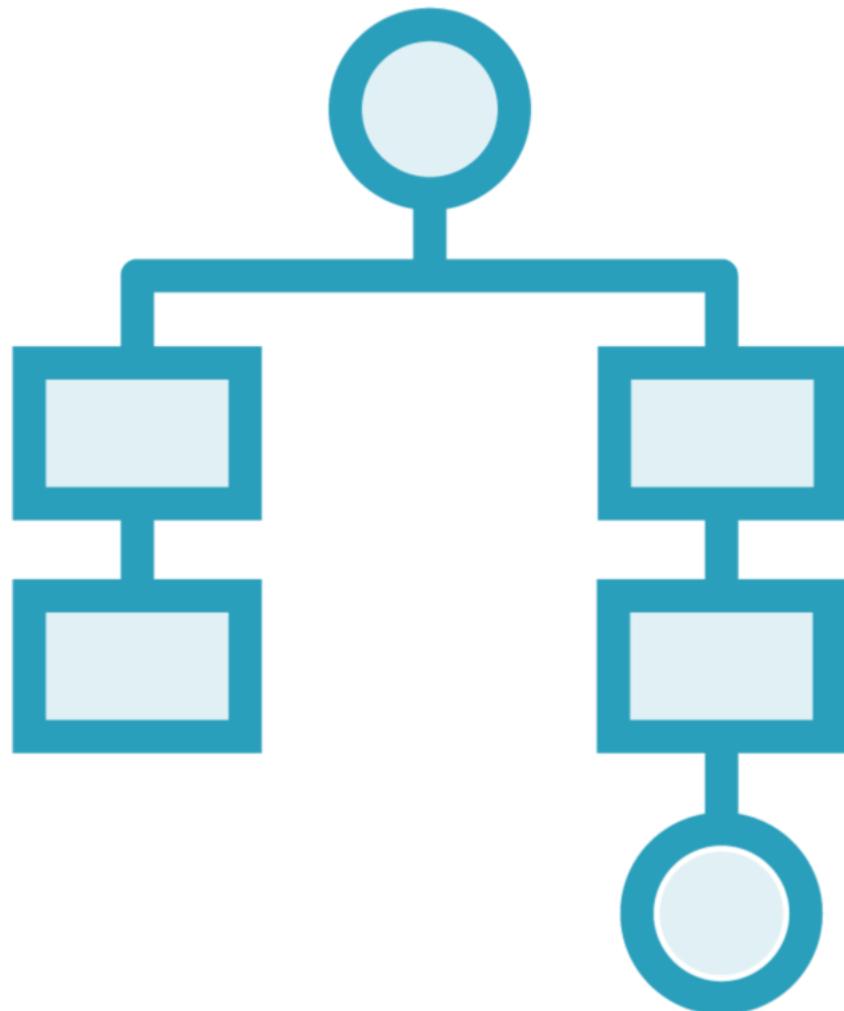
Lift state up if non-child component needs the data



Next choice: Context or external state management



useContext Hook



Context is designed to share data that is global for a tree of React components.

Avoid passing props through multiple layers of components.

Examples: Theme, demographic information, language, etc..



Easy setup with Create React Apps



Create React App

```
npx create-react-app project
```



Lesson 3: TypeScript is a life saver!



Why TypeScript?

**Catch problems
early on**

**Intellisense is
accurate**

**Easier to refactor
code**

Readable code

**Easier to maintain
and test**

**High quality
documentation
(TSDoc)**



```
interface MessageProps {  
  text: string;  
  important: boolean;  
}  
  
export const Message = ({ text, important }: MessageProps) => {  
  return (  
    <div>  
      {important ? "Important message: " : "Regular message: "}  
      {text}  
    </div>  
  );  
};
```

Component in TypeScript

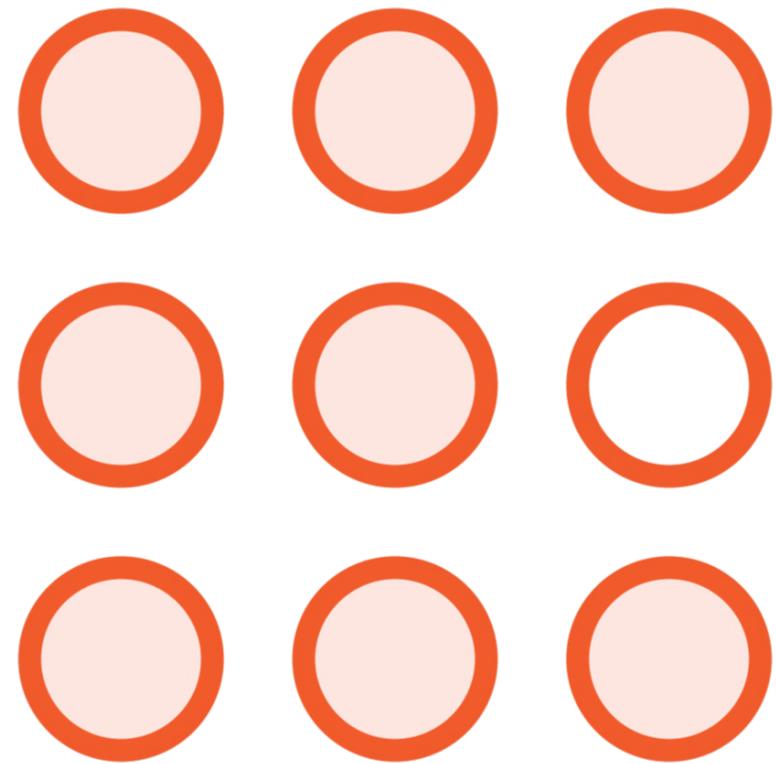
Interface defines the *props* that are accepted by the component using an object type

MessageProps is the interface that describes the *props* the component accepts

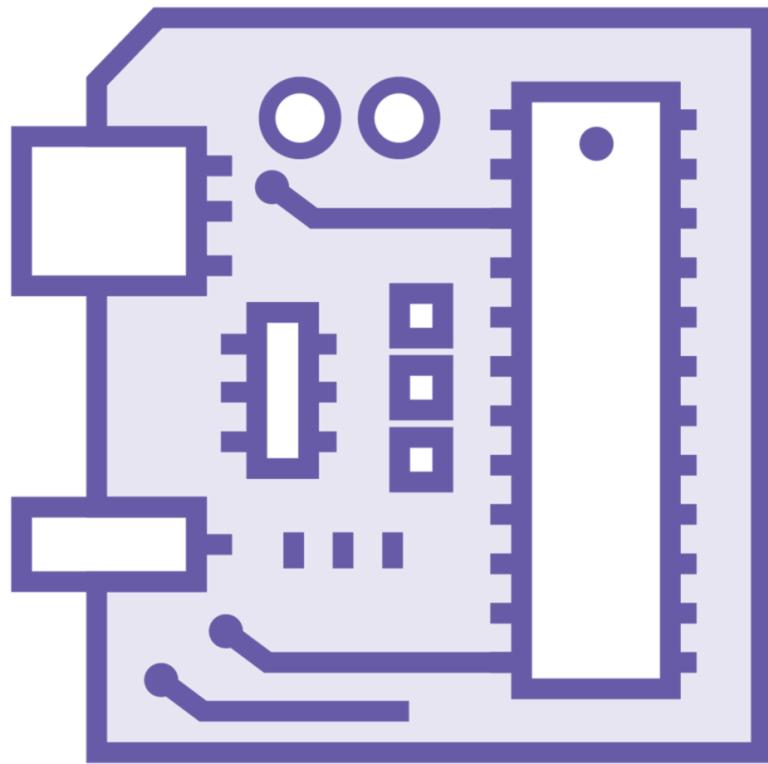
Test, Test and Test!



Types of Testing



Unit testing - Jest



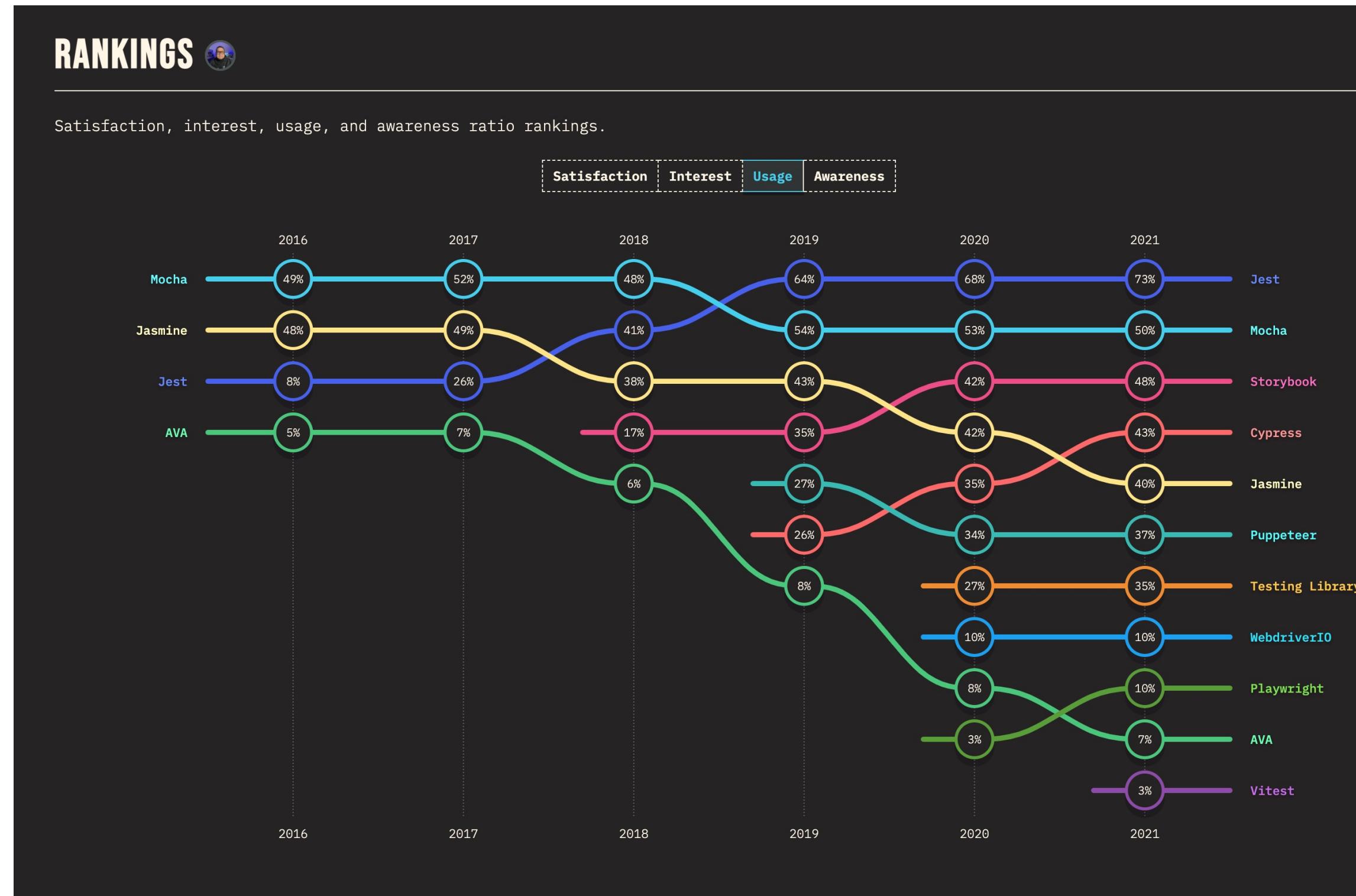
**Component testing -
React testing library**

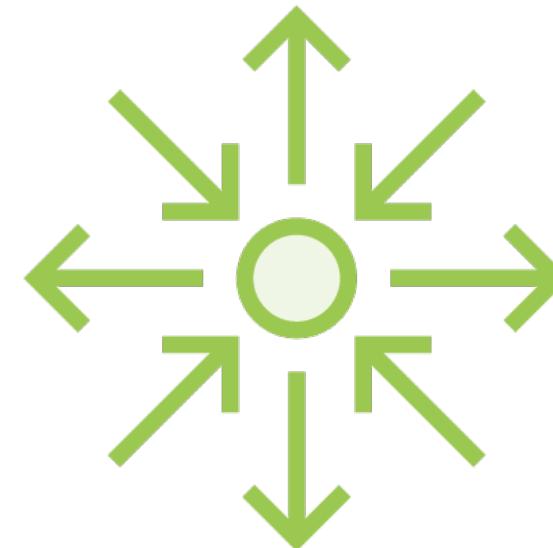
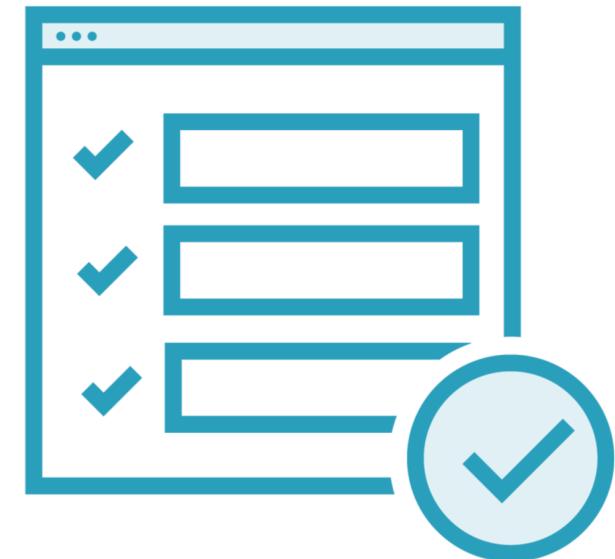
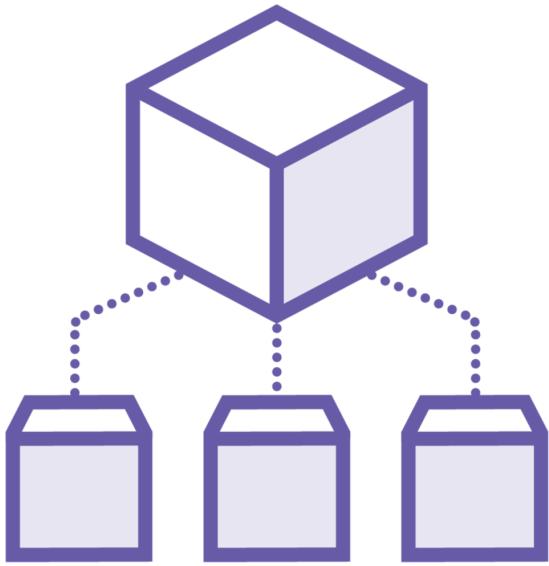


**Automated End-to-
end tests - Cypress**



State of JS Survey 2021





Developer Decisions



- Routing**
- Data fetching**
- Server-side rendering**
- Search Engine Optimization**
- Image Optimization**
- Bundling**
- Code Splitting**
- Deployment**



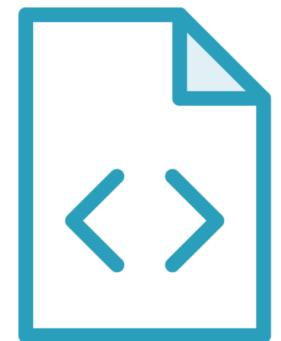
What is Next.js?



Next.js is a flexible React framework that gives you building blocks to create fast web applications



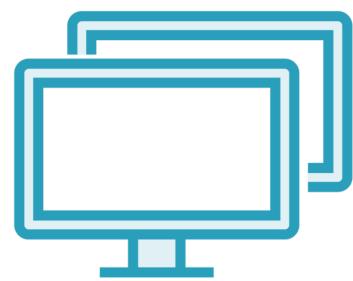
Next Features



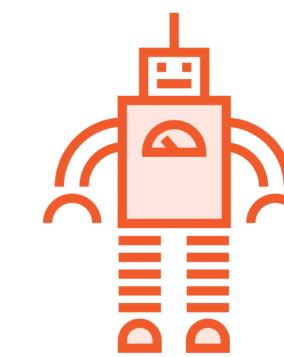
File-system Routing



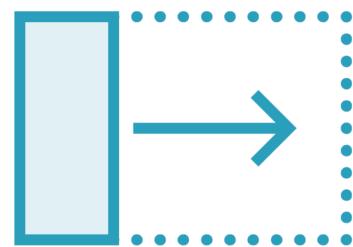
Image and font optimization



Hybrid: SSG and SSR



Automatic bundle splitting and lazy loading



Incremental Static Regeneration



Automatic TypeScript support



Next.js is a React Framework for Production



Thank You!





CONTENT AUTHORED

7

All time

Adhithi Ravichandran

Pluralsight Author

Following

439 Followers

Adhithi Ravichandran is a Software Consultant, Author and Speaker based in Kansas City. She is the owner and founder of Surya Consulting, Inc. through which she provides her expertise in Software Architecture, Development and Training. She provides clients, consulting services in architecting...

[Show more...](#)[@adhithiravichandran.com](mailto:adhithiravichandran.com)[Twitter](#)[LinkedIn](#)

TOPICS AUTHORED



GraphQL

Content authored

Cypress 9 Fundamentals

Course · Beginner · 1 hr 59m · May 23, 2022 · ★★★★★ (24)

Understanding Culture Intelligence

Course · Intermediate · 55m · Jul 11, 2021 · ★★★★★ (15)

React Native: The Big Picture

Course · Beginner · 1 hr 3m · Jan 27, 2021 · ★★★★★ (194)

React Native 0.63: Components Playbook

Course · Intermediate · 2 hr 31m · Jan 12, 2021 · ★★★★★ (31)

Consuming a GraphQL API with Apollo Client 3 and React

Course · Intermediate · 2 hr 1m · Sep 23, 2020 · ★★★★★ (79)

Cypress 4: End-to-end JavaScript Testing

Course · Intermediate · 2 hr 9m · May 5, 2020 · ★★★★★ (254)

GraphQL: The Big Picture

Course · Beginner · 1 hr 17m · Aug 7, 2019 · ★★★★★ (672)

Checkout my courses on Pluralsight on React Native, GraphQL, Cypress, etc..



@AdhithiRavi

