

React Basics



Tooling up

- TypeScript
- WebPack
- React

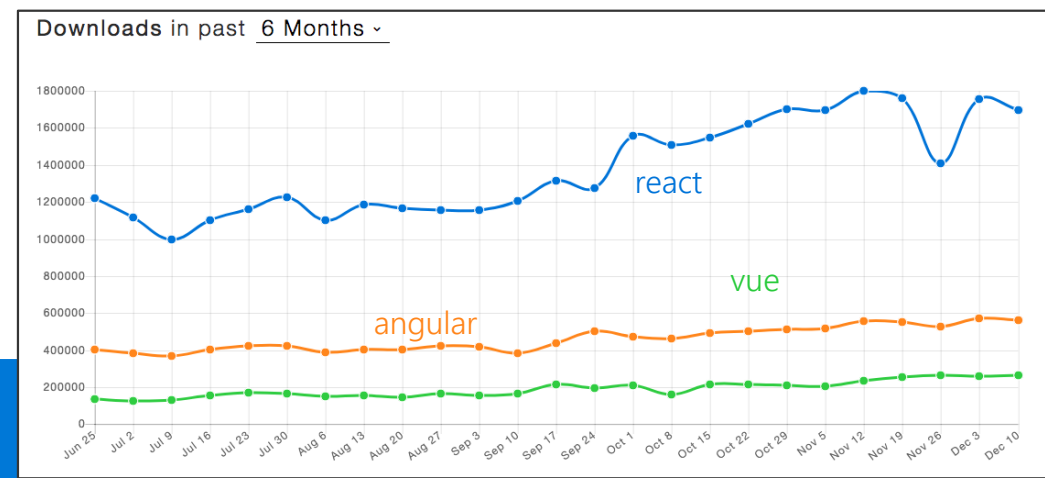







Getting Started

- First SPFx Project
- Accessing SharePoint and Graph APIs
- Provisioning
- Web part Editing
- Extensions



Popular UI libraries



	Library or Framework	Source	Notes
	AngularJS 165K	Google	Complete framework designed for building SPAs MV* pattern; strong dependency injection; many services built in Angular 2+ not yet able to run multiple instances – bad for web parts ☹
	React 139K	Facebook	UI library Anti-MVVM, HTML in the JavaScript JSX/TSX format adds HTML into your code
	Vue 58.8K	Evan You (crowd-sourced sponsorship)	UI library MVVM pattern – bind JavaScript in the HTML
	Knockout 59K	Steve Sanderson (MS employee – but KO is not a MS product)	UI library; designed for pages but works for components too MVVM – binding attributes in HTML, observable objects in JS Older – works with more browsers
	jQuery 85K	John Resig	DOM manipulation library with added features (AJAX, promises ...) Designed to provide consistency among browsers Older – works with more browsers

Why React?

- SPFx is written in React ... it's already on the page
- Faster than AngularJS
- Office UI Fabric Components
- Property Panel Controls
- Designed for component-ness

Why NOT React?

- Not opinionated, so you might have to make **lots** of decisions (don't worry ... SPFx makes them for you)
- Angular's Dependency Injection is pretty sweet
- Somewhat steep learning curve
- You hate JSX/TSX, and/or you have HTML designers who can't deal with it
- State management can be a challenge in large/complex env'ts (add a state manager like Redux or MobX)

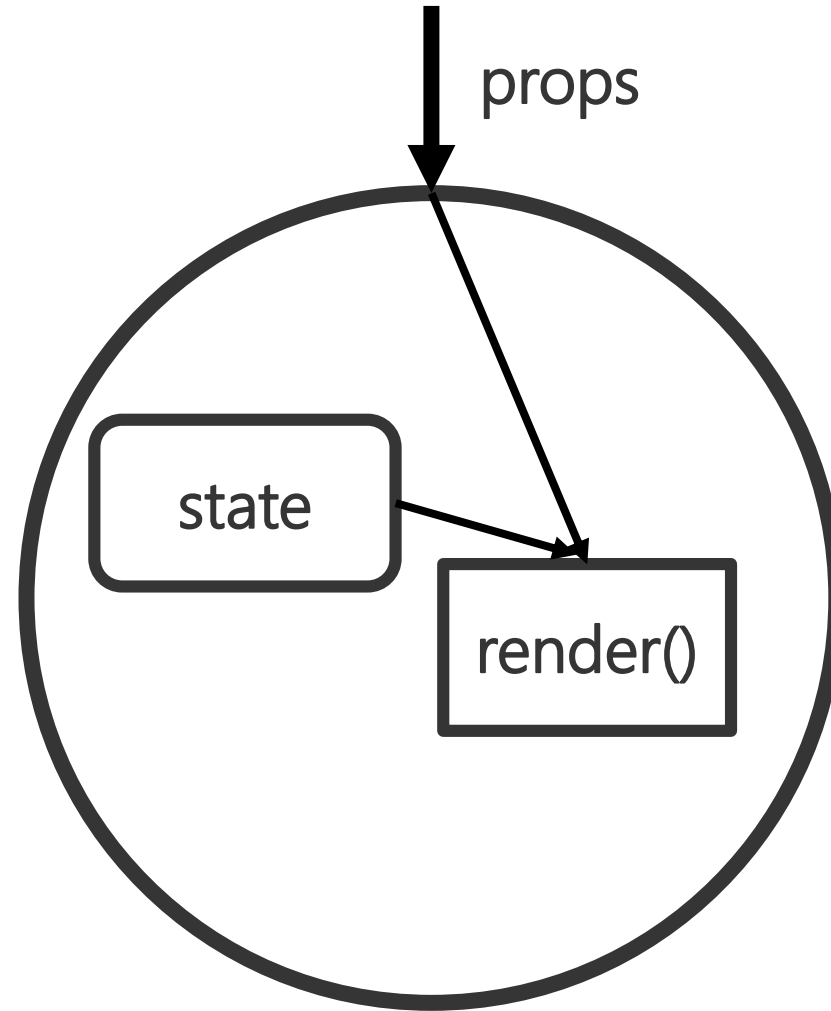
Quiz time!

```
let a=1;  
let b=2;  
let c=a+b;  
b = 100;  
console.log(c);    // What is logged?
```

Components

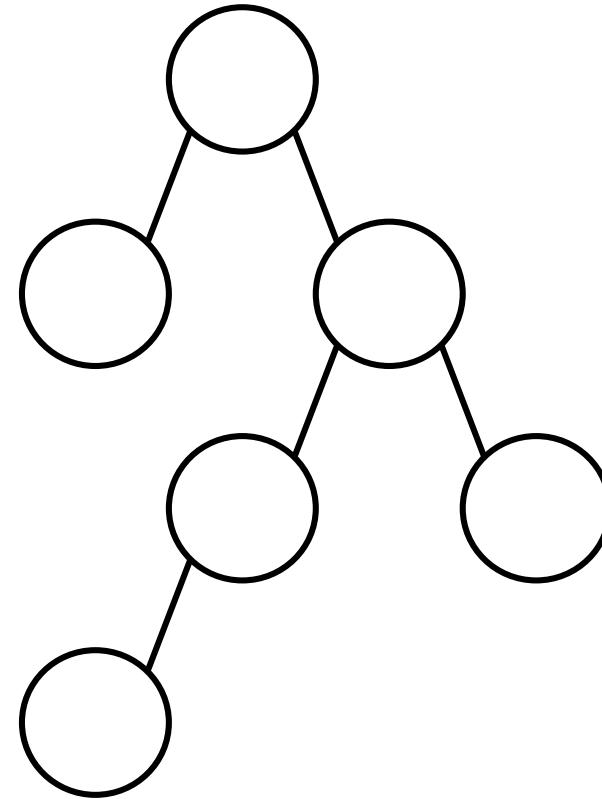
Every component:

- Can have properties (props)
Props are 1-way; they only change when parent renders
- Can have state
State is persistent and survives re-renders
- Must have a render() method
Render creates child components and sets their props



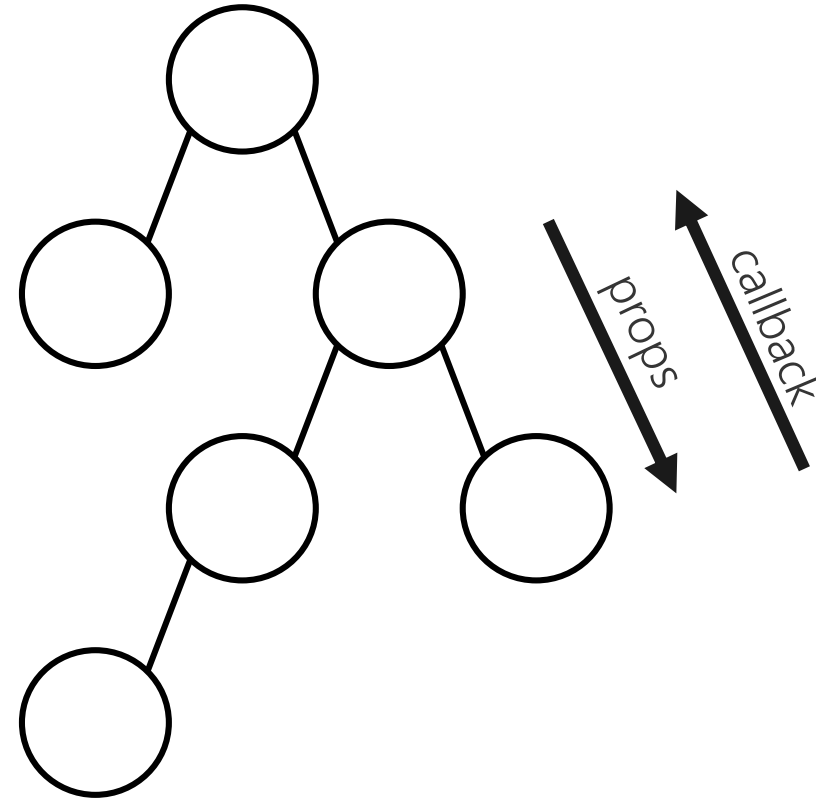
Components

- There is a built-in component for every HTML element
- There are also component libraries and components you write
- A hierarchy of components can render a widget or a whole page



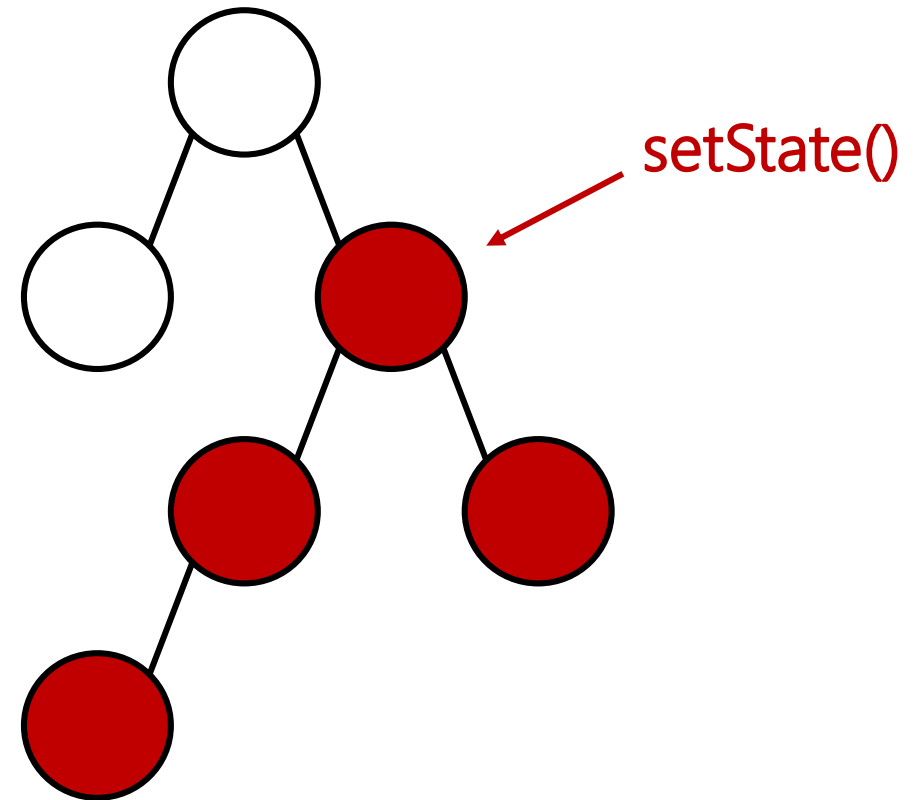
Components

- Communication from parent to child is by passing new properties during render()
- Communication from child to parent is by a callback function passed to the child in a property



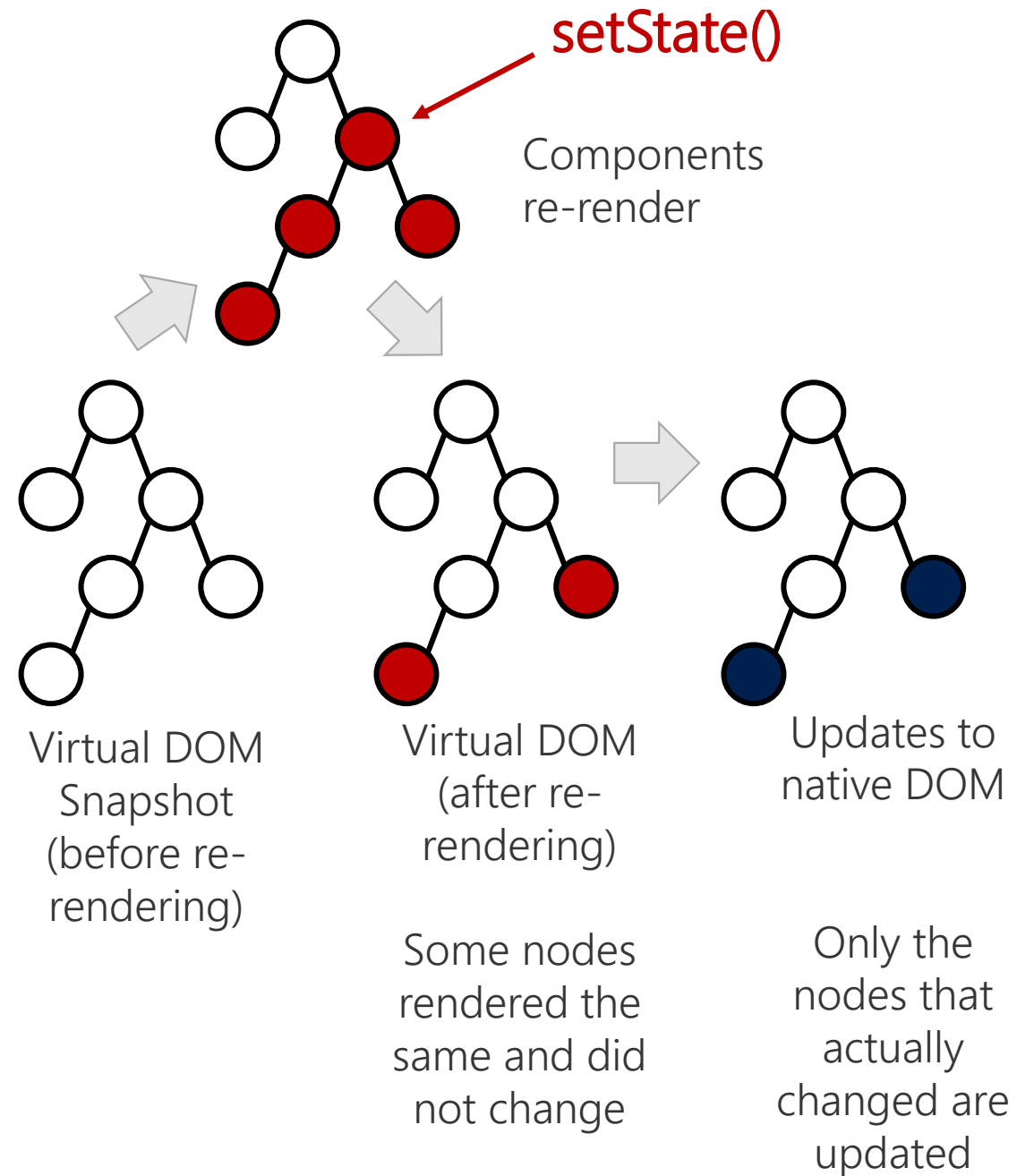
Components

- Components update their state with the `setState()` method
- When state is set, by default that and all child components re-render
- Based on updated state and properties, a component could render different children, or set the children's properties differently

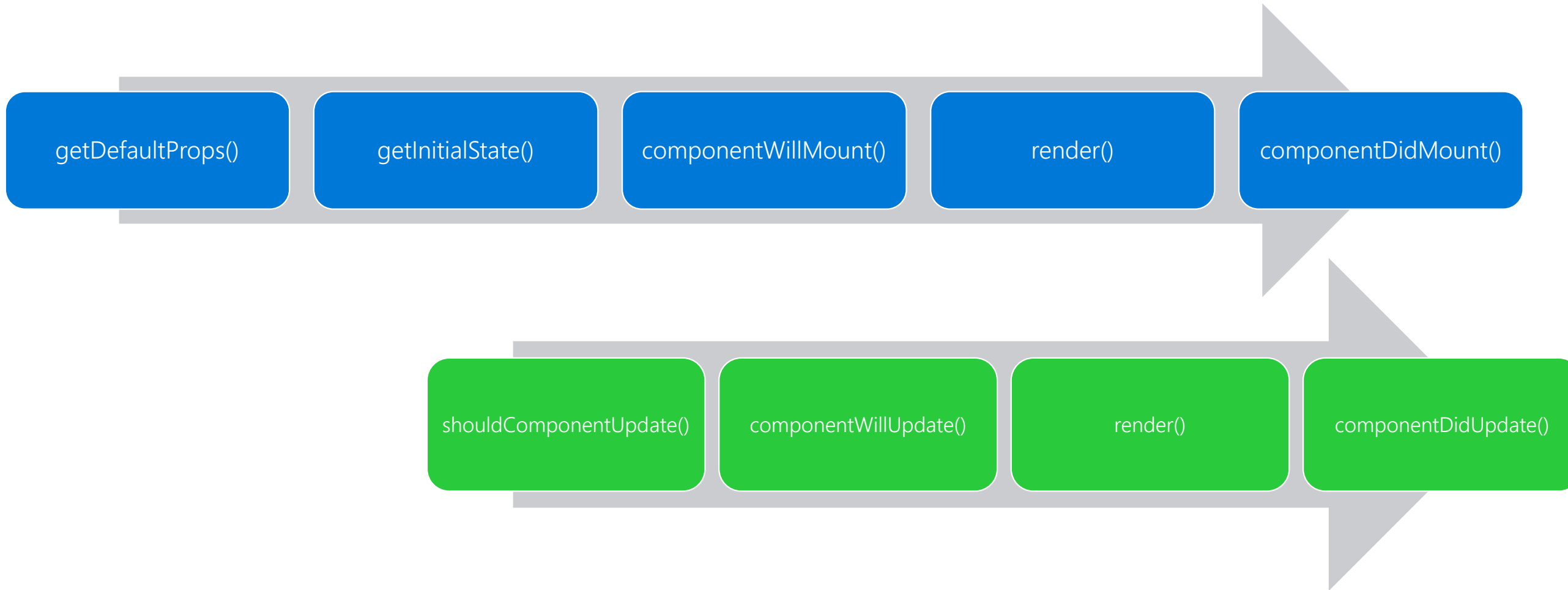


Virtual DOM

- When components re-render, React updates the *virtual DOM*
- When that's done, React very efficiently updates the *native DOM* with only the changes



React Lifecycle



JSX/TSX

```
public render(): void {  
  const element =  
    React.DOM.div({  
      children: [  
        React.createElement(  
          ComponentA,  
          { url: this.state.url } ),  
        React.createElement(  
          ComponentB,  
          { url: this.state.url } ),  
      ]  
    });  
  return, this.domElement);  
}
```

```
public render(): void {  
  const element =  
    <div>  
      <ComponentA url={this.state.url} />  
      <ComponentB url={this.state.url} />  
    </div>;  
  return, this.domElement);  
}
```

Do try this at home!
<https://link.bobg.tv/BabelJSX>

JSX/TSX

1. Common parent
2. Proper XML syntax
3. No quotes around tags
4. Lower case tags are HTML components
5. Capitalized tags are custom components
6. {} to put JavaScript expression in attributes or child

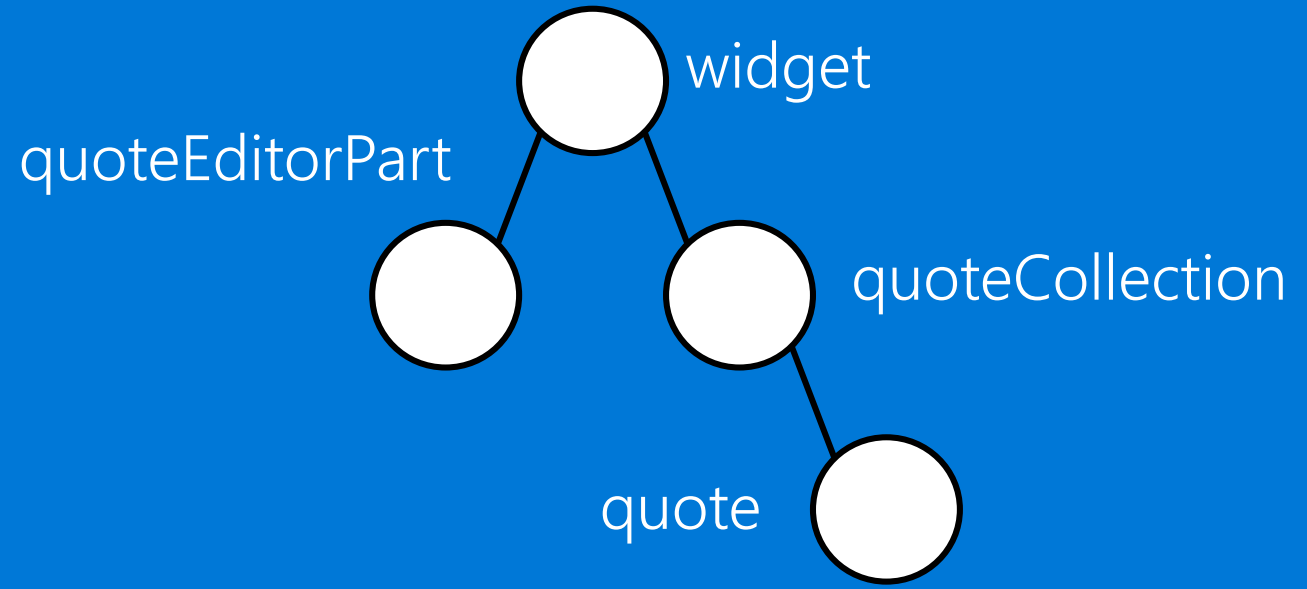
```
public render() {
  return (
    <div className={styles.quotes}>
      {this.state.quotes.map(q =>(
        <QuoteDisplay quote={q} />
      ))}
      <CommandButton
        className={styles.lastLine}
        icon='Refresh'
        onClick={this.handleClick.bind(this)}>
        {this.props.getMoreLabel}
      </CommandButton>
    </div>
  );
}
```

Instant Lab

React with JavaScript and jQuery

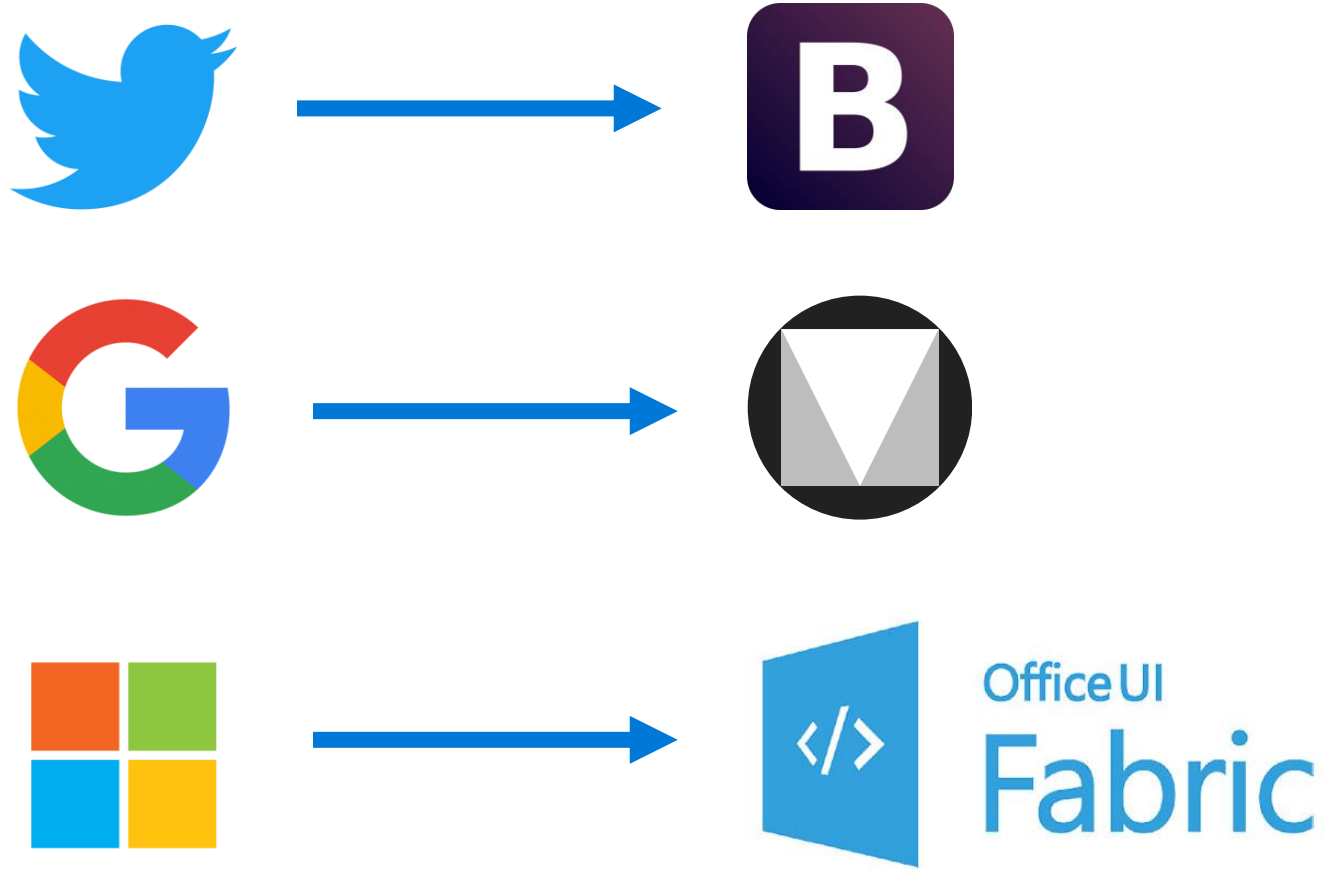
Shows:

- Simple component hierarchy
- JSX/TSX Syntax
- Updates based on user input



<https://bit.ly/ReactQuotes>

Responsive Design Language Frameworks



Instant Lab

Office UI Fabric Components

<http://bit.ly/OfficeFabric>

Resources

React Site

<https://facebook.github.io/react/>

React Parts Catalog

<https://react.parts>

Facebook React Tutorial

<http://bit.ly/FB-React-Tut>

Redux Site

<http://redux.js.org/>

