



FRONTEND DEVELOPMENT WITH REACT

QCC TECH WORKS

INTRODUCTION TO STATE AND PROPS

- ▶ Goals:
 - ▶ Understand what State and Props are.
 - ▶ Gain a deeper knowledge of JSX, and Javascript (ES2015 syntax).
 - ▶ Learn to change State
 - ▶ Learn to pass, and use props in components.



RECAP ON STATE

- ▶ Data is called "state"
- ▶ More precisely "state" is data that changes
- ▶ State is always an object with key value pairs
- ▶ Is always initialized in the constructor

```
constructor() {  
  super();  
  this.state = {  
    name: 'React'  
  };  
}
```

HOW TO CHANGE STATE

- ▶ React has a built in method to change state called `setState()`
 - ▶ When the `setState()` method is executed the `render()` method is executed as well.
 - ▶ These events cause the DOM to be updated
- ▶ When you update "state", React re-renders the view for you
- ▶ React re-renders in a performant way. By using a complex "diffing" algorithm.

```
onClick={()=>this.setState({name: "Jesse"})};
```

PROPS

Think of props as “options” passed to a component to customize its functionality.

Props are conceptually and syntactically very similar to an HTML property.

All Props are passed into a component become key-value pairs on that component’s “props” object.

```
render() {  
  return (  
    <div>  
      <Hello name={this.state.name} />  
      <p>  
        Start editing to see some magic happen :)  
      </p>  
    </div>  
  );  
}
```

```
1  
2  
3  const Hello = (props) => {  
4    return (  
5      <h1>Hello {props.name}</h1>  
6    )  
7  };  
8  
9  export default Hello  
10
```

RENDER METHOD

Think of the render method as your template

- Called whenever the state changes
- Here we decide what the user should see based on the state
- Notice that even though we are calling render on every state change, not everything is getting reloaded on the page. The entire DOM is not re-rendering

```
render() {  
  return (  
    <div>  
      <Hello name={this.state.name} />  
      <p>  
        Start editing to see some magic happen :)  
      </p>  
    </div>  
  );  
}
```

VIRTUAL DOM

Think of the virtual DOM as React's local and simplified copy of the HTML DOM. It allows React to do its computations within this abstract world and skip the “real” DOM operations, often slow and browser-specific.

—React KungFu