State and Props

Anish Kumar

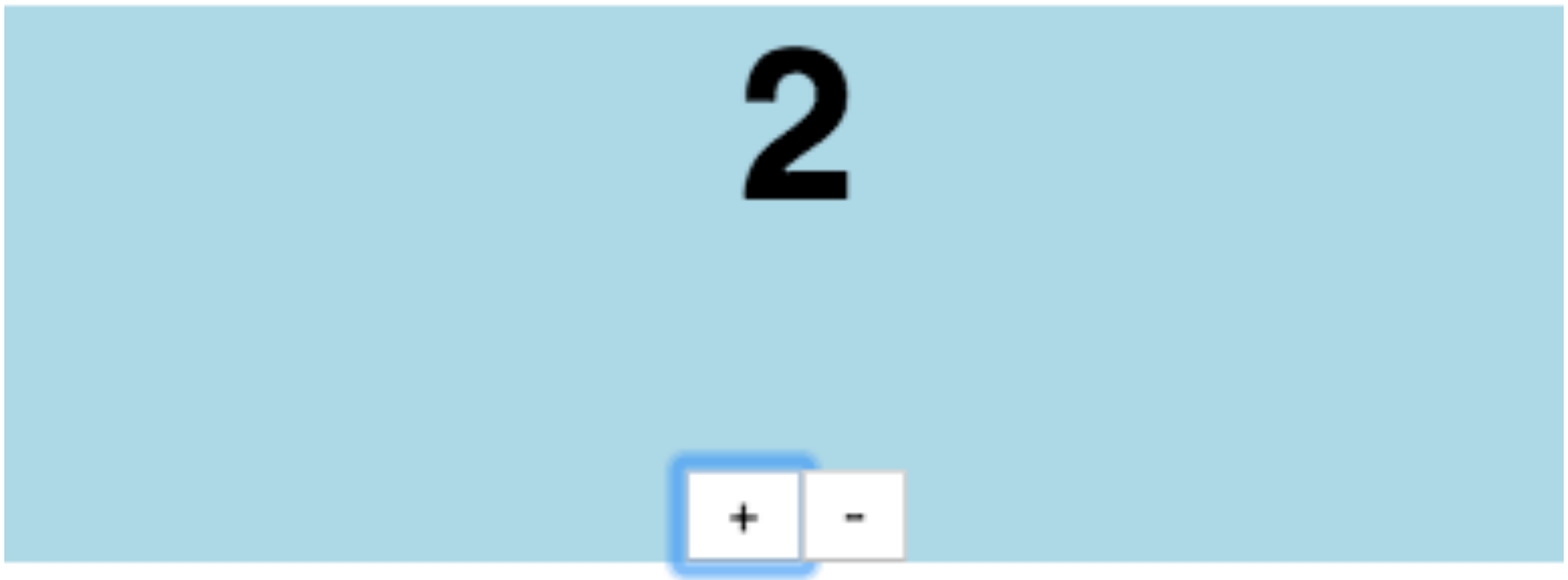
Software Developer at Grappus

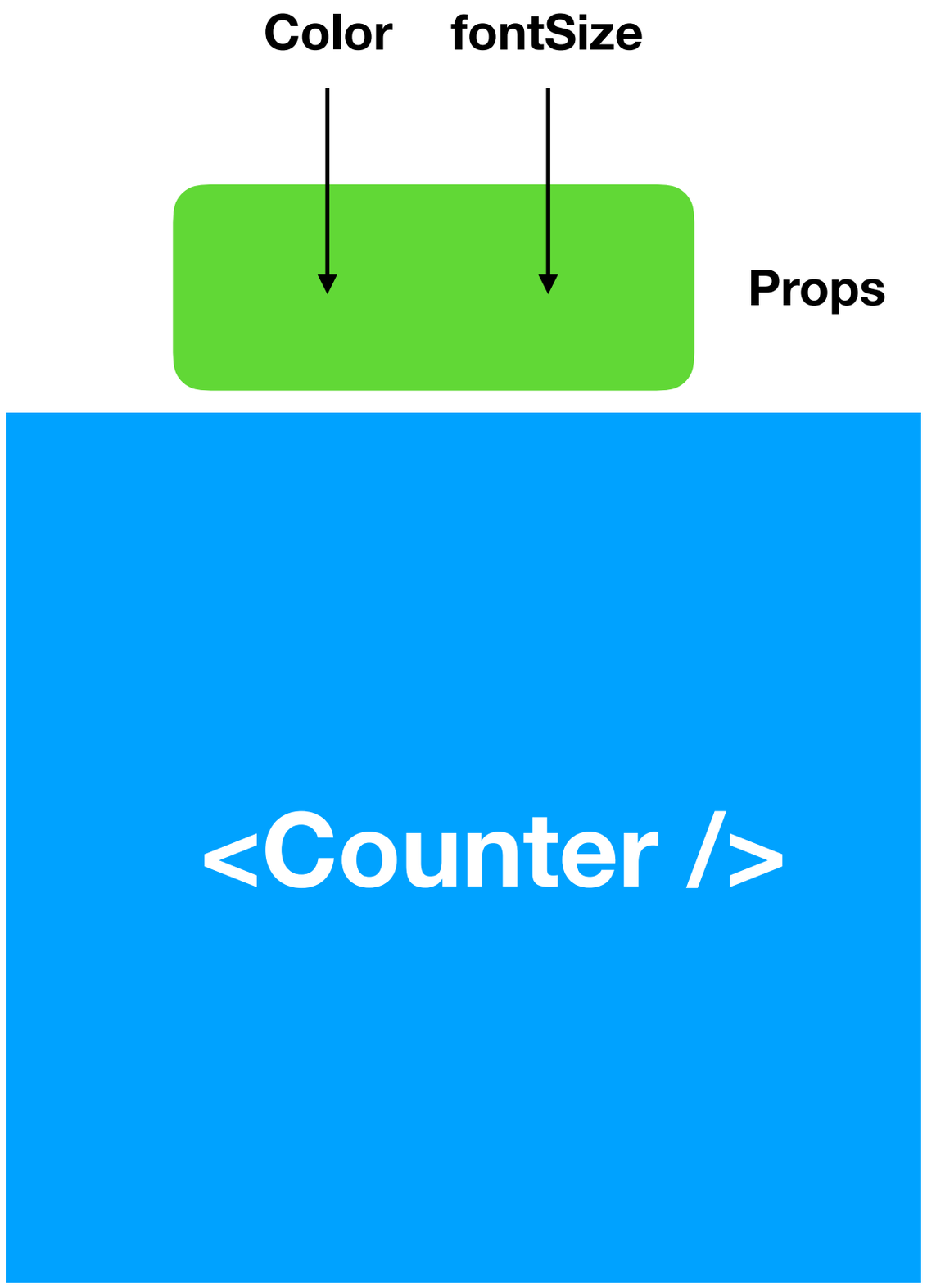
@_anish_kr

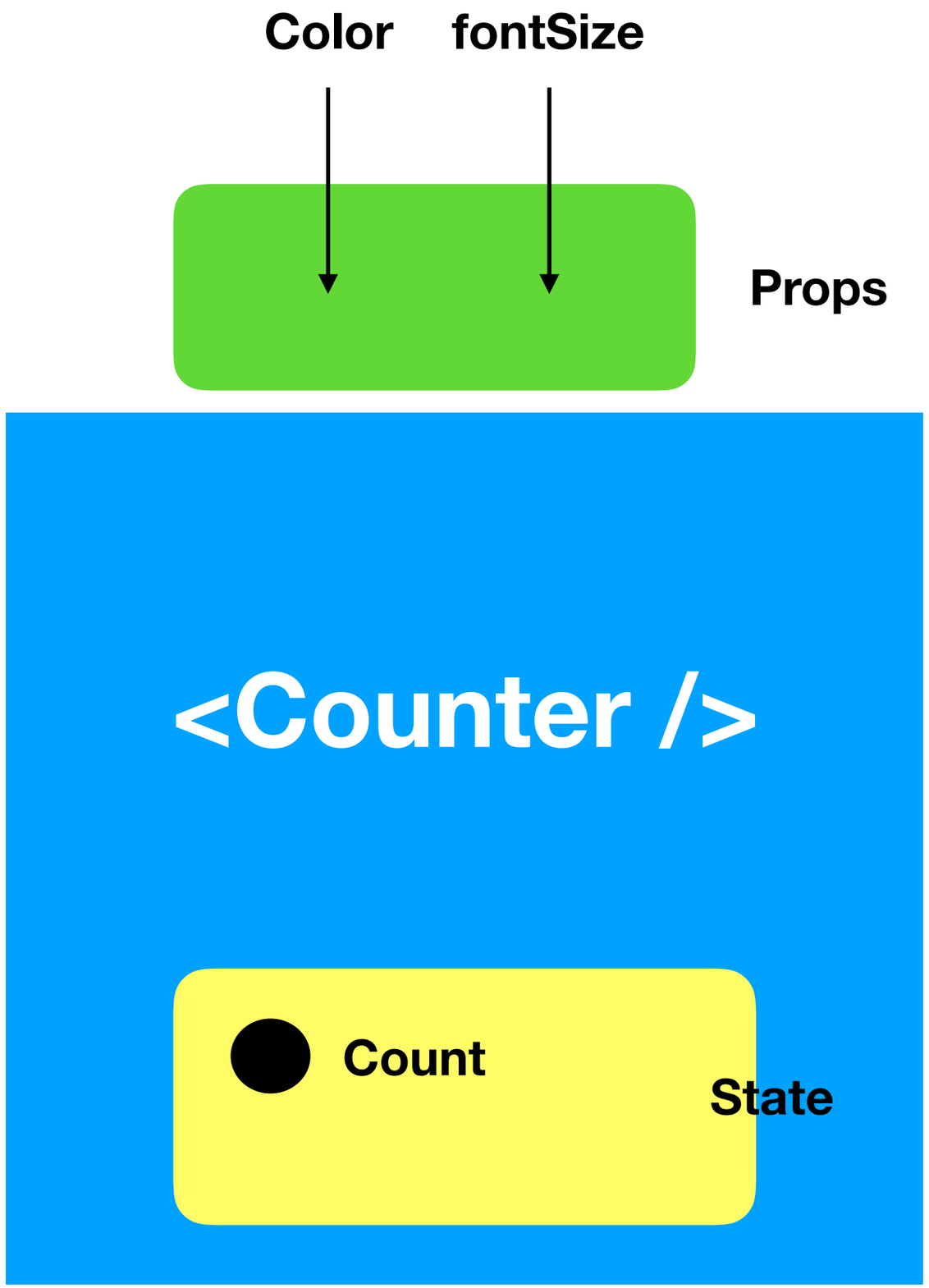
anish000kumar

A simple component

< Counter />




Color
fontSize
Props
<Counter />


Color
fontSize
Props
<Counter />
Count
State

Props

```
// component definition
class Welcome extends React.Component {
  render() {
    return <h1>Hello {this.props.name}</h1>;
  }
}
```

```
// HomePage.js
const Home = () =>(

  <div className=“main”>
     <Welcome name="Sara" />
  </div>

);
```

Props 101:

Props “come from above” (Parent)

Props should not change (Immutable)

Props can be assigned default values (defaultProps)

Props can be assigned types (propTypes)

PropTypes are checked in development mode only

Props can be any of these:
string / number / boolean / Array /
Object / function / React Component

(Any valid JS object/type basically :D)

State

Quick Example

```
class Welcome extends React.Component {

  state= {
    visitorCount: 0
  }

  render() {
    return(
      <h1>Hello {this.props.name}</h1>
      <p>
         You are visitor number {this.state.visitorCount}
      </p>
    )
  }
}
```

State 101

State can be mutated
(just a fancy way of saying it can be changed)

State can be should be initialised with some default value

State should be changed via **setState** method

Every time **setState** changes state, the component re-renders

setState

```
2     state = {
3       name: "",
4       age: 0,
5     }
6
7     this.setState({
8         name: "Anish Kumar"
9     })
```

```
1
2    state = {
3        name: "",
4        age: 0
5    }
6
7    setState( state => ({
8            name: "Anish Kumar",
9            id: 54
10       })
11   )
```

https://codesandbox.io/s/4zwo21qo99

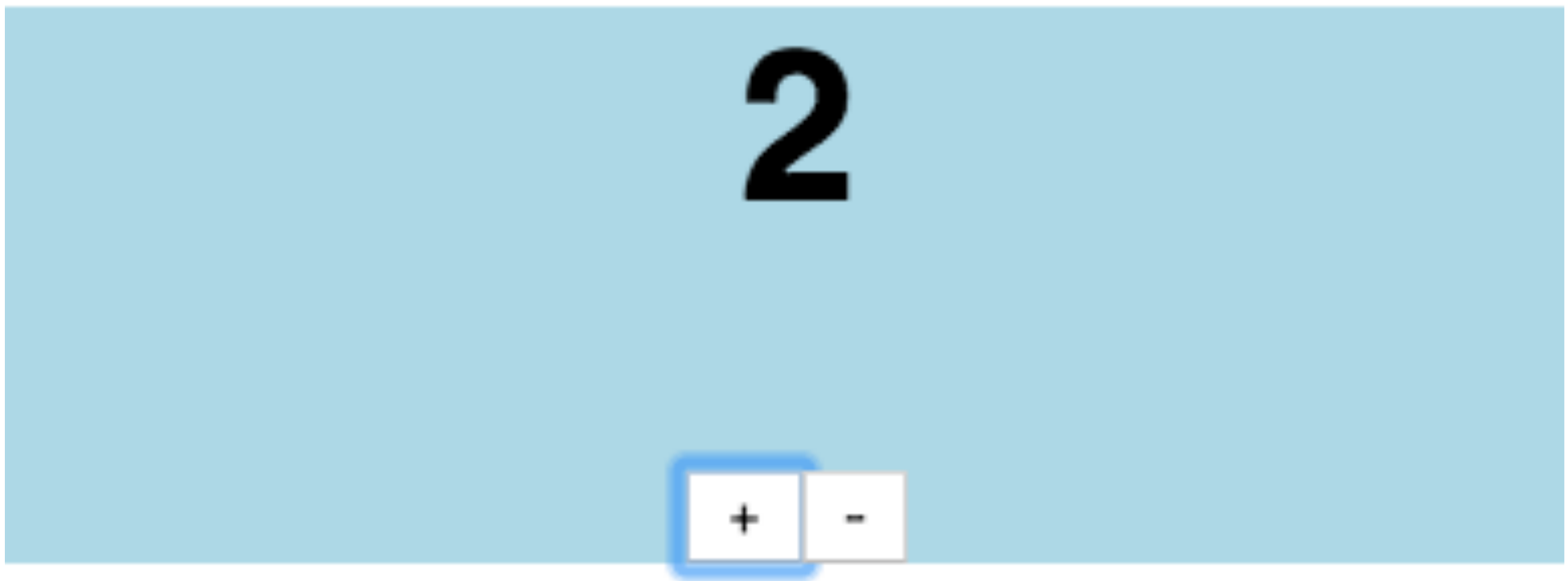
```
1
2   // setState is asynchronous
3
4   state = {
5       count: 1
6   }
7
8   this.setState({
9       count: this.state.count+1
10  })
11  this.setState({
12      count: this.state.count+1
13  })
```

```
1
2   // setState accepts a callback too
3
4       const callback = () => {
5         alert('Hey! setState called me :D')
6       }
7
8     this.setState({
9         count: this.state.count+1
10    }, callback)
```

Let’s finish the counter component

https://codesandbox.io/s/n3194202j

```
46    <Counter color="lightblue" fontSize={42} />
```

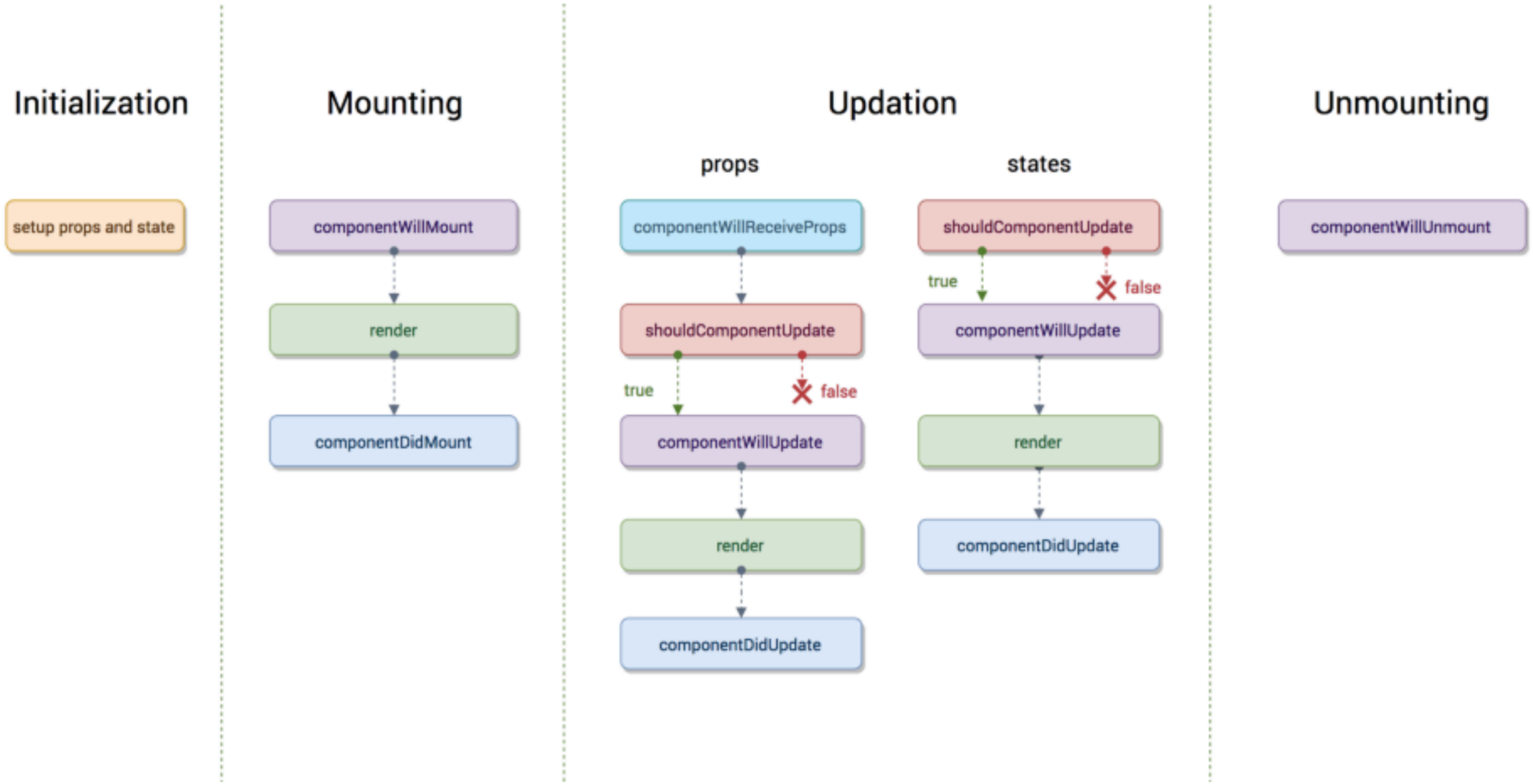


```
4   export default class Counter extends React.Component {
5     static defaultProps = {
6       color: "black",
7       fontSize: 16
8     };
9
10    static propTypes = {
11      color: propTypes.string,
12      fontSize: propTypes.number
13    };
14
15    state = {
16      count: 0
17    };
```

```
31      render() {
32        return (
33          <div style={{ background: this.props.color }} className="counter">
34            <h3 style={{ fontSize: this.props.fontSize + "px" }}>
35              {this.state.count}
36            </h3>
37
38            <button onClick={this.countUp}> + </button>
39            <button onClick={this.countDown}> - </button>
40          </div>
41        );
42      }
```

```
19      countUp = () => {
20        this.setState({
21          count: this.state.count + 1
22        });
23      };
24
25      countDown = () => {
26        this.setState({
27          count: this.state.count - 1
28        });
29      };
```

LifeCycle Hooks



Queries?

Thank You