
CSE 331

Software Design & Implementation

Topic: More React

 **Discussion:** Why are the keys on a keyboard not aligned properly?

Reminders

- **Strongly** recommend going to section tomorrow
- No late days on HW9

Upcoming Deadlines

- HW7 due Thursday (8/03)

Last Time...

- Review
- Examples
 - Simplest React application
 - Character Counter

Today's Agenda

- Review
- Examples
 - Class Picker
 - Messaging App

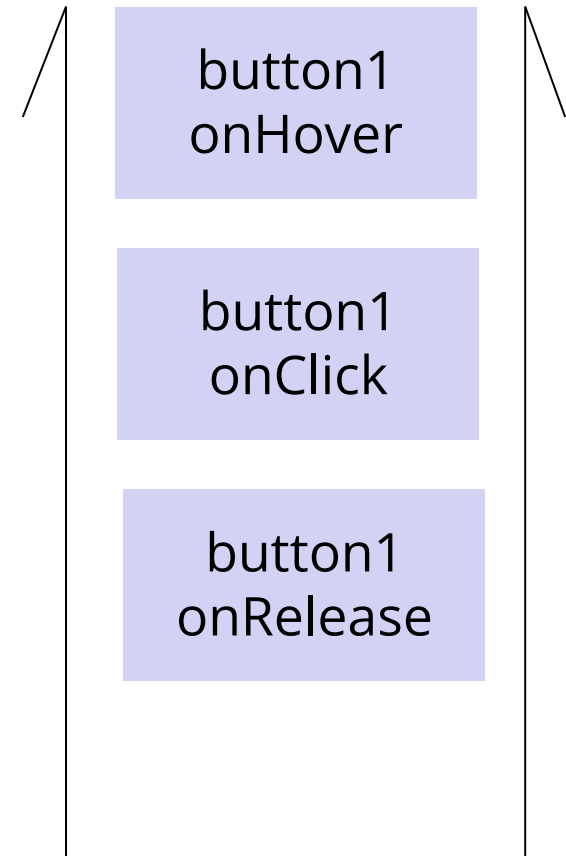
Event-driven programming

Register Event

```
public void myFunction() {  
    System.out.println("I was here");  
}  
button1.addOnClickListener(myFunction);
```

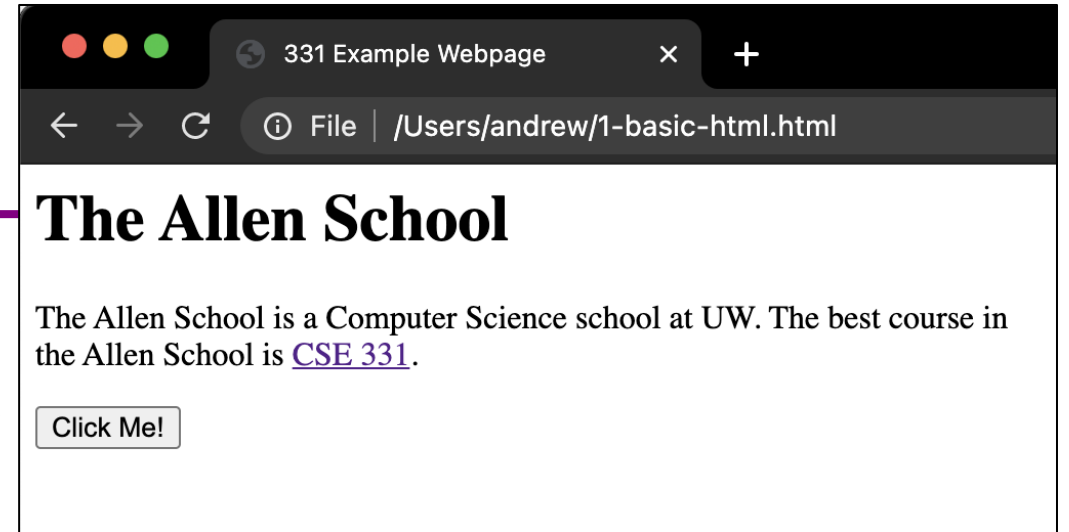
Event loop:

```
do {  
    e = getNextEvent();  
    process event e;  
} while (e != quit);
```



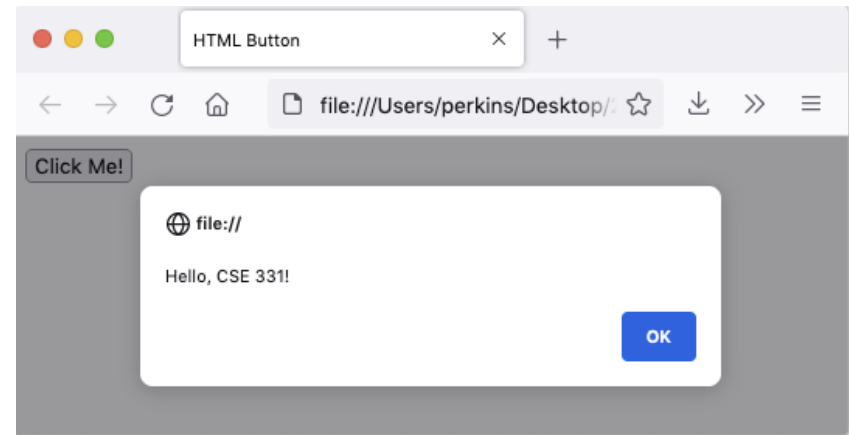
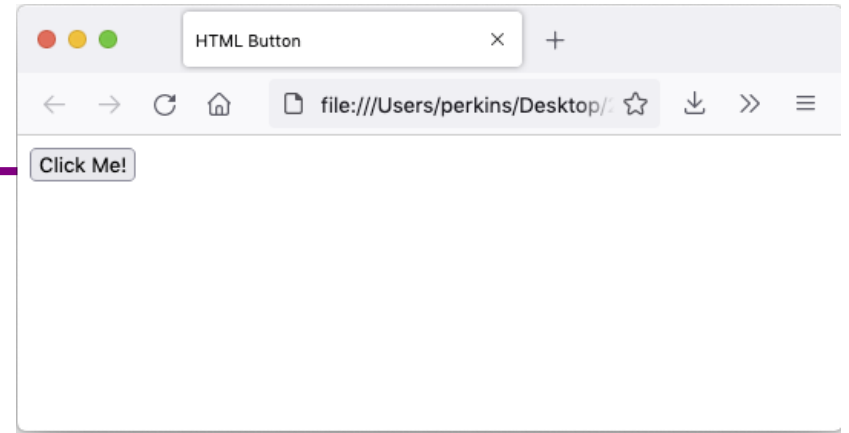
Demo

```
<html lang="en">
  <head>
    <title>331 Example Webpage</title>
  </head>
  <body>
    <h1>The Allen School</h1>
    <div>
      <p>
        The Allen School is a Computer Science school at
        UW. The best course in <br/> the Allen School is
        <a href="https://cs.uw.edu/331">CSE 331</a>.
      </p>
      <button>Click Me!</button>
    </div>
  </body>
</html>
```



Demo 2

```
<html lang="en">
  <head>
    <title>HTML Button</title>
  </head>
  <body>
    <script type="text/javascript">
      function sayHello() {
        alert("Hello, CSE 331!");
      }
    </script>
    <button onclick="sayHello()">Click Me!</button>
  </body>
</html>
```



Reminder: Our Stack

(we write these)



Compiled/Combined by
the Development Tooling



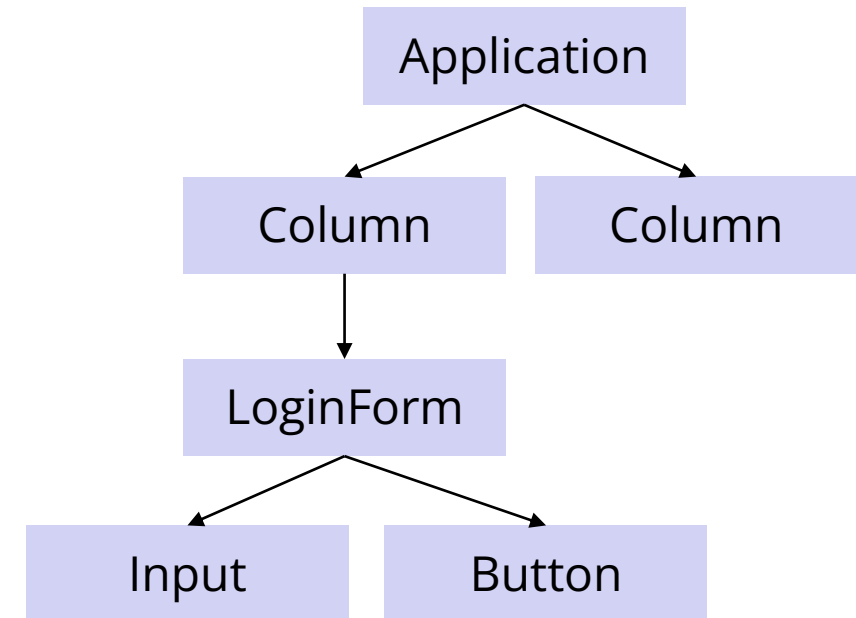
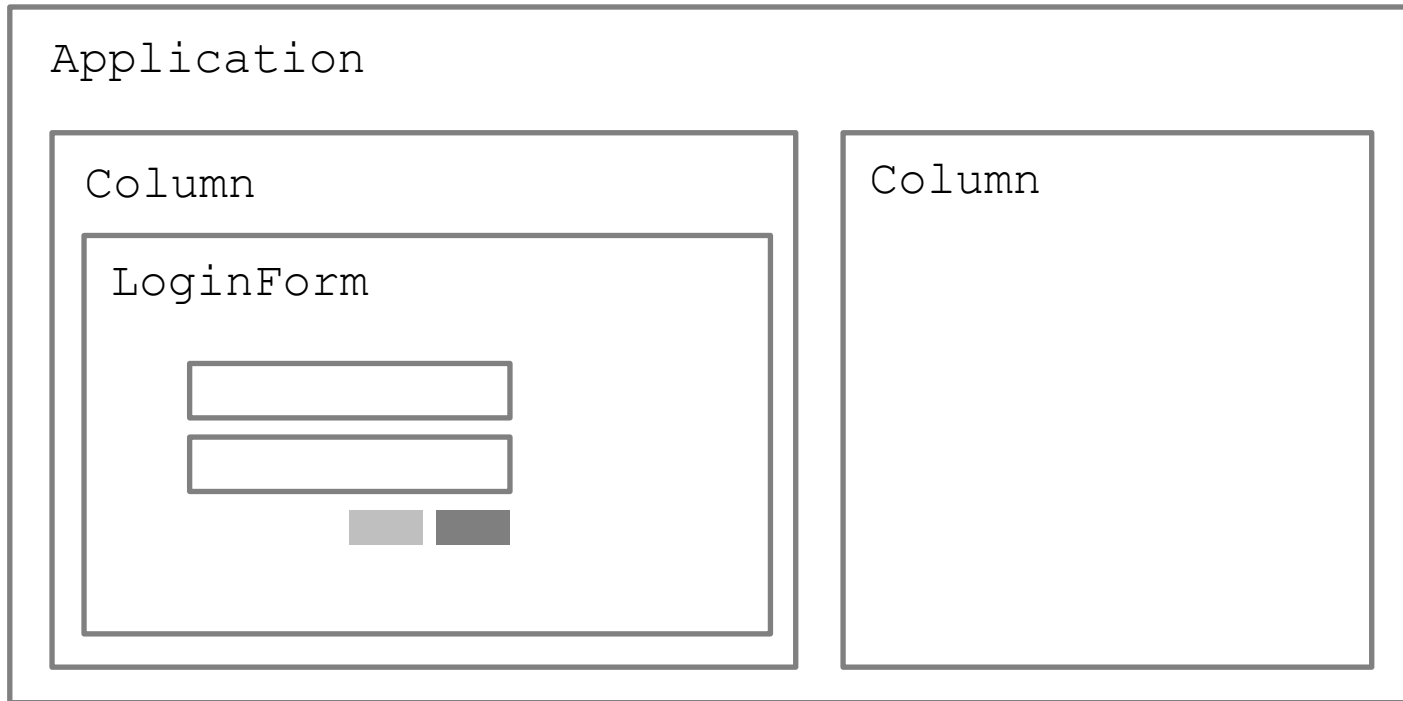
(sent to browser to execute)

Example 1

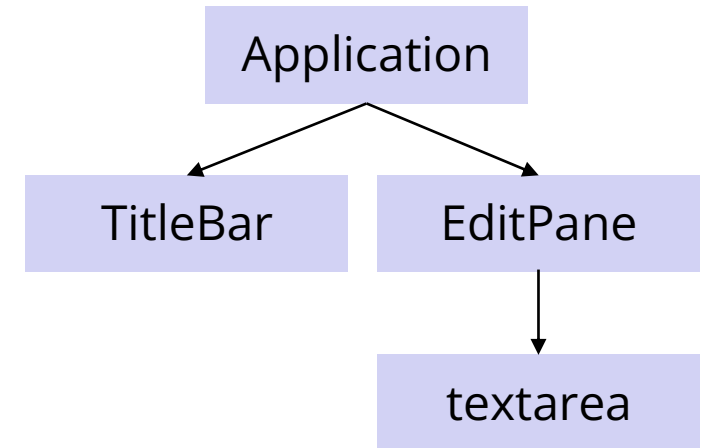
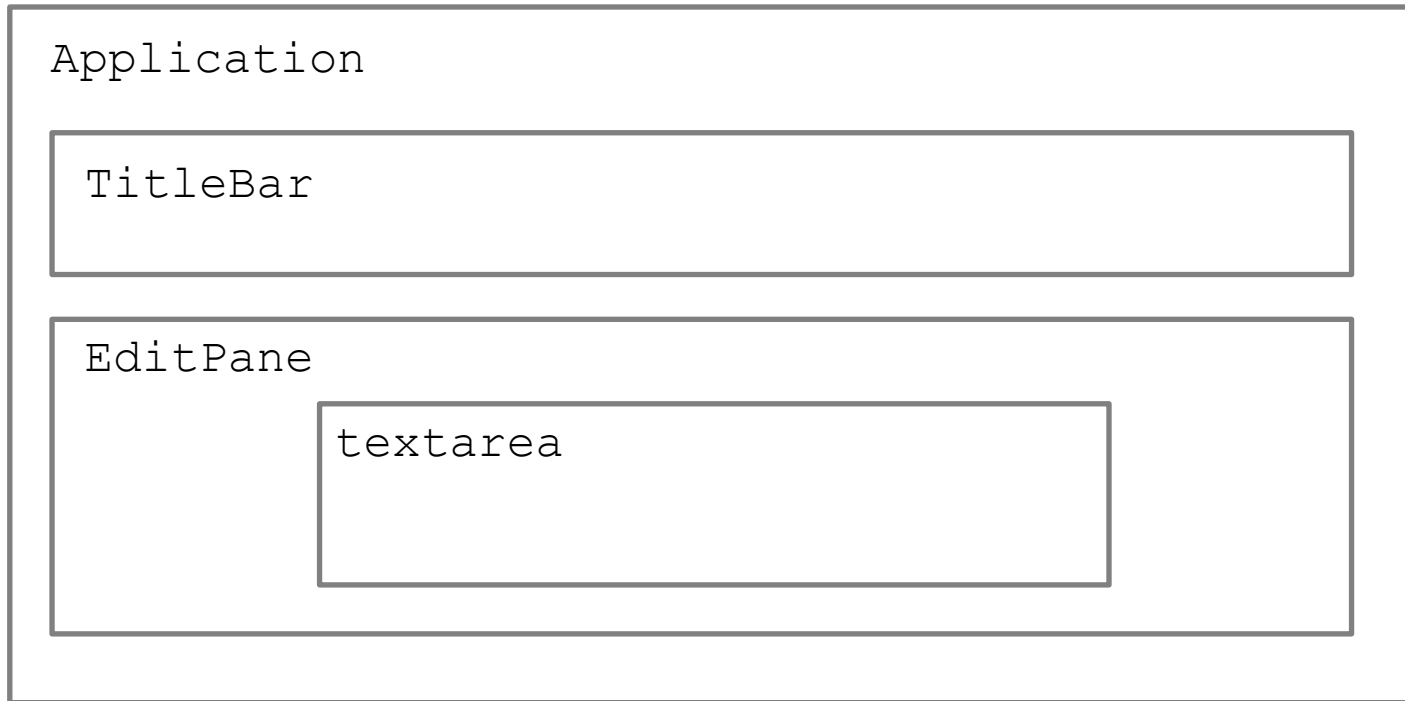
- The simplest source code to create a React website is these 3 files:
 - `index.html`
 - A very small amount of "necessary" HTML
 - Most of the actual web content will be generated by the TS/React code
 - `index.tsx`
 - Starting point of code – runs when the page loads
 - Starts React
 - `App.tsx`
 - Our first component – the App component
- When we build the React app, all these files will be incorporated into what is sent to the browser

Components

- We will have many components
 - e.g. Application, Column, LoginForm, Input, Button



Components

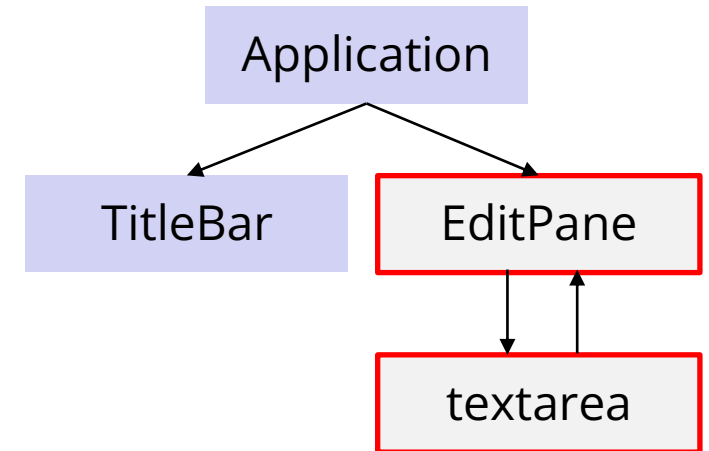
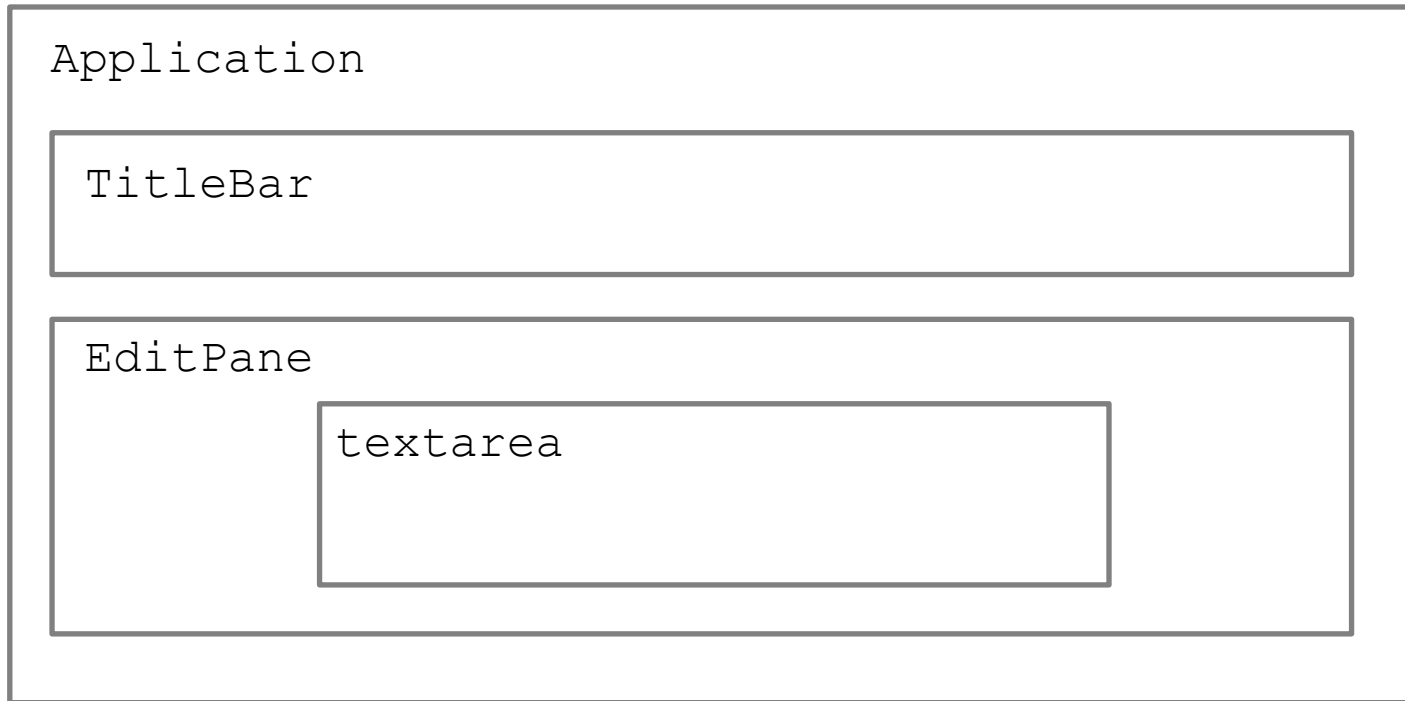


Example 3

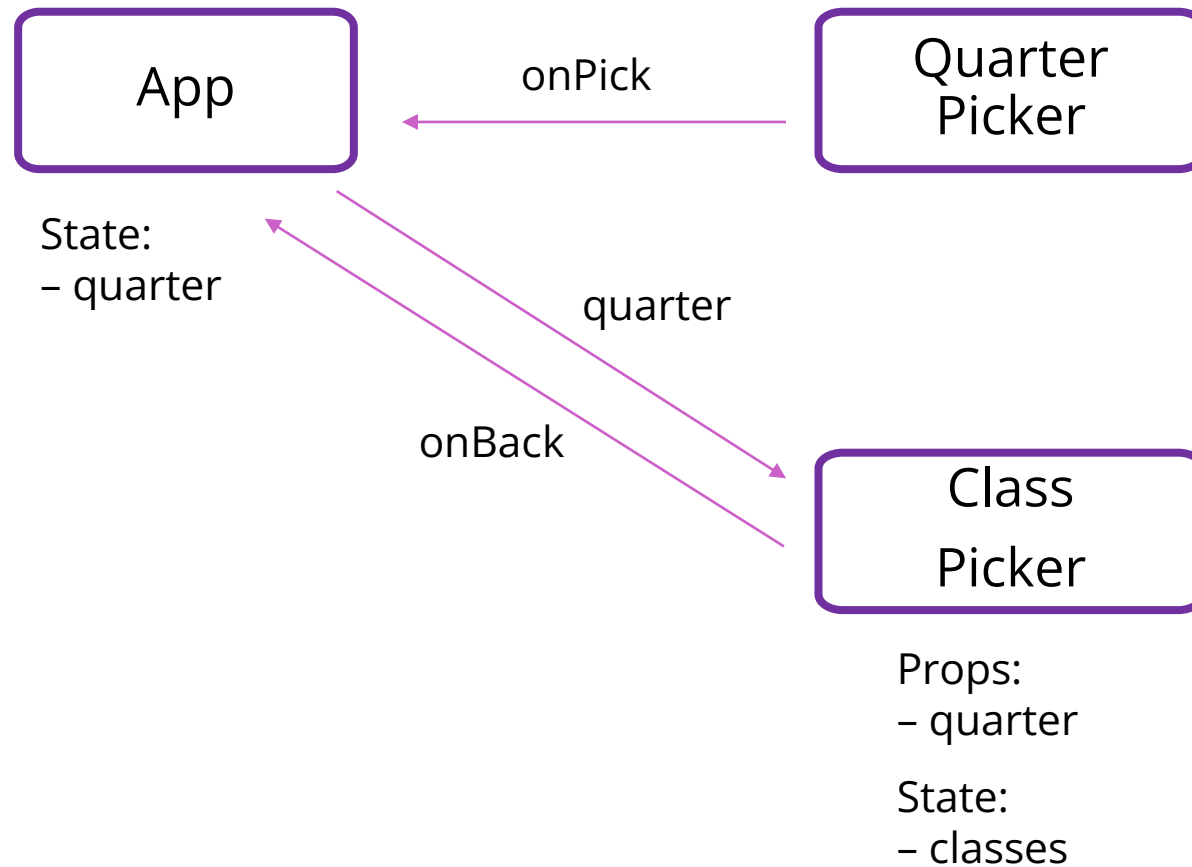
register-react2/...

Passing Data from Parent -> Child

- We saw some data passed between components in Example 2
 - e.g. Application uses props to pass data to TitleBar, EditPane



Structure of Example React App



React setState

- `setState` does not update state instantly:

```
// this.state.x is 2
this.setState({x: 3});
console.log(this.state.x); // still 2!
```

- Update occurs after the event finishes processing
 - `setState` adds a new event to the queue
 - work is performed when that event is processed
- React can batch together multiple updates

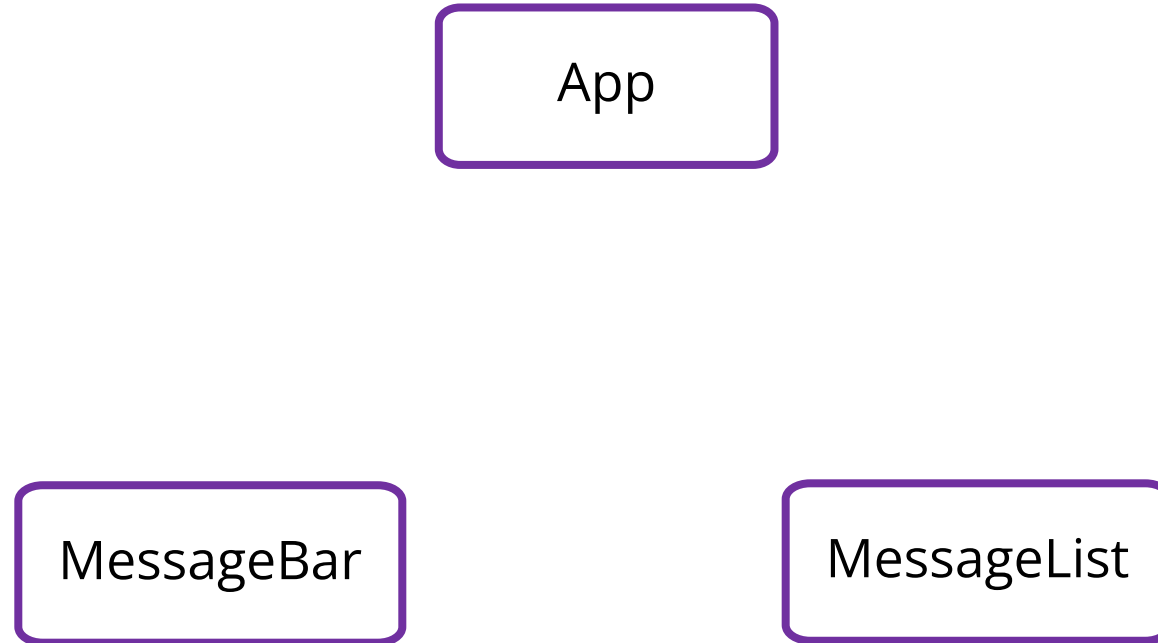
React Gotchas

- `render` should not have side-effects
 - only *read* `this.state` in render
- Never modify `this.state`
 - use `this.setState` instead
- Never modify `this.props`
 - read-only information about parent's state
- Not following these rules may introduce bugs that will be hard to catch!

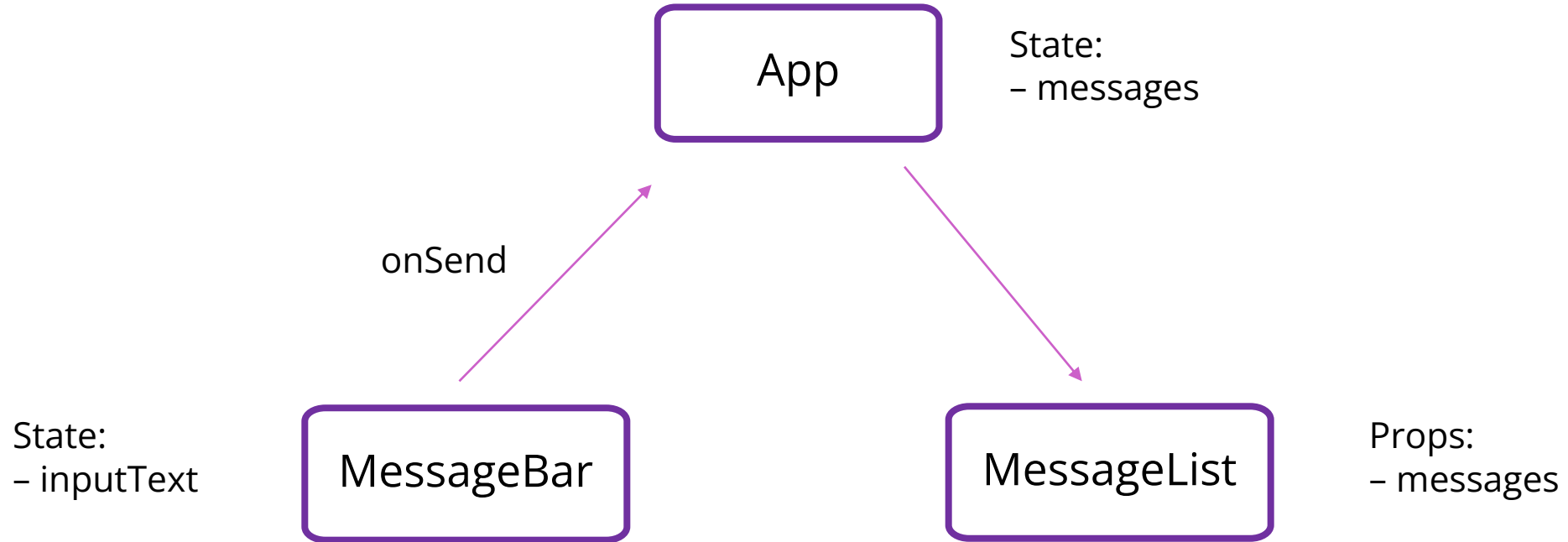
Example 4

messaging/...

How should data flow?



Data Flow



Before next class...

1. Ask questions about [HW7](#) if you have any
 - Will need to apply generics
 - Useful for implementing Dijkstra's algorithm on a **Graph<Double>**