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 (8/04)

Last Time…

- Review
- Examples
  - Simplest React application
  - Character Counter

Today's Agenda

- Review
- Examples
  - Class Picker
  - Messaging App

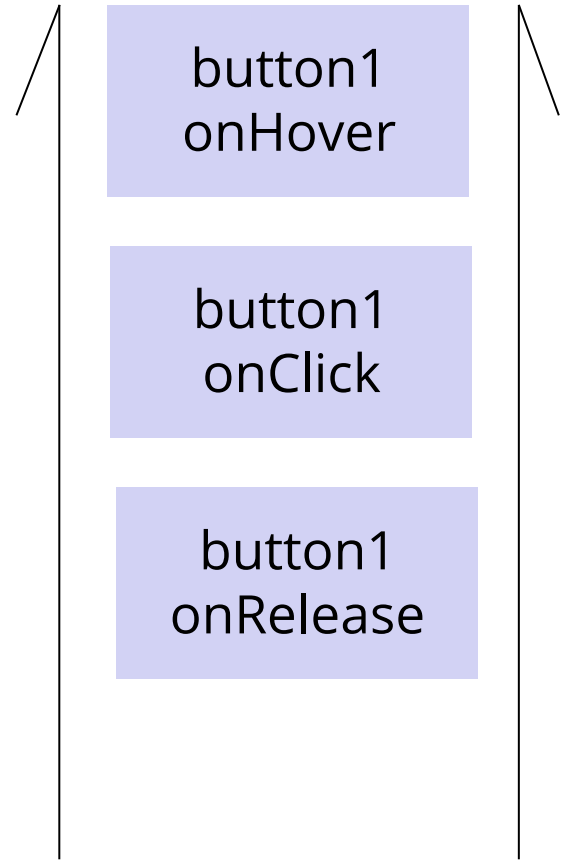
Event-driven programming

Register Event

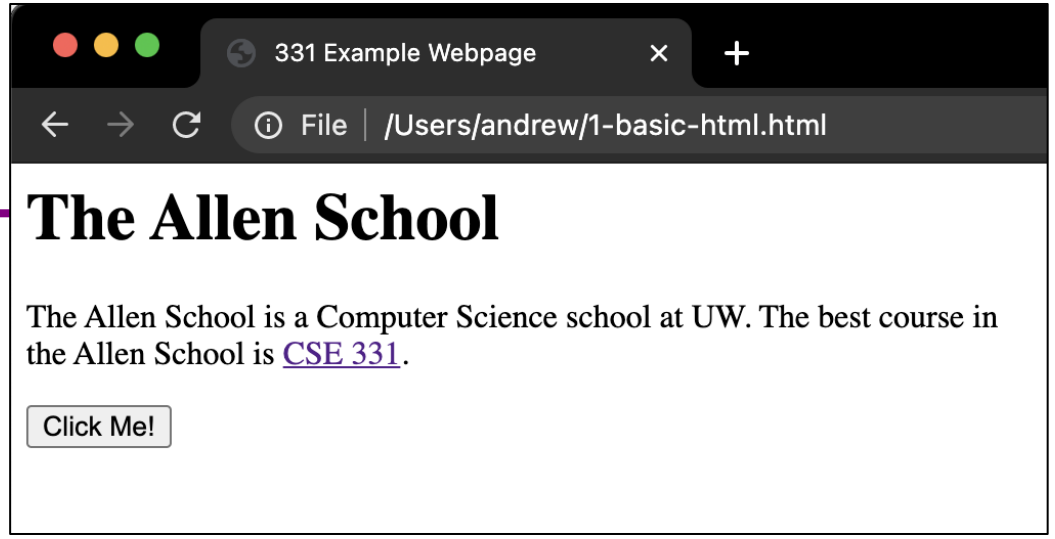
```
public void myFunction() {
    System.out.println("I was here");
}
button1.addOnClickListener(myFunction);
```

Event loop:

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



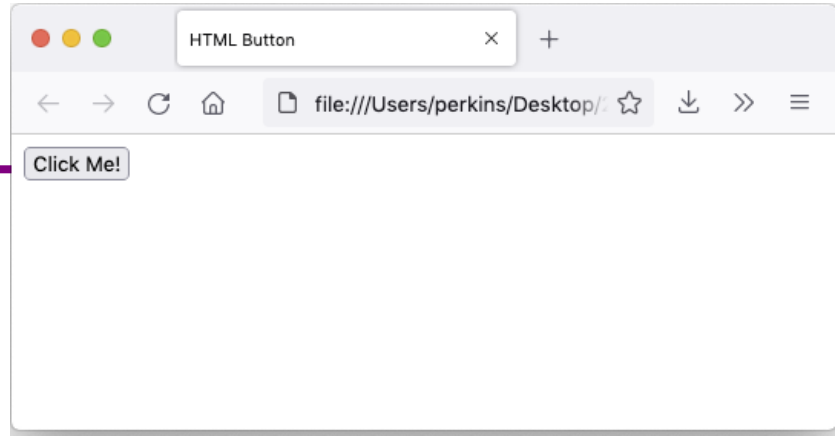
Demo

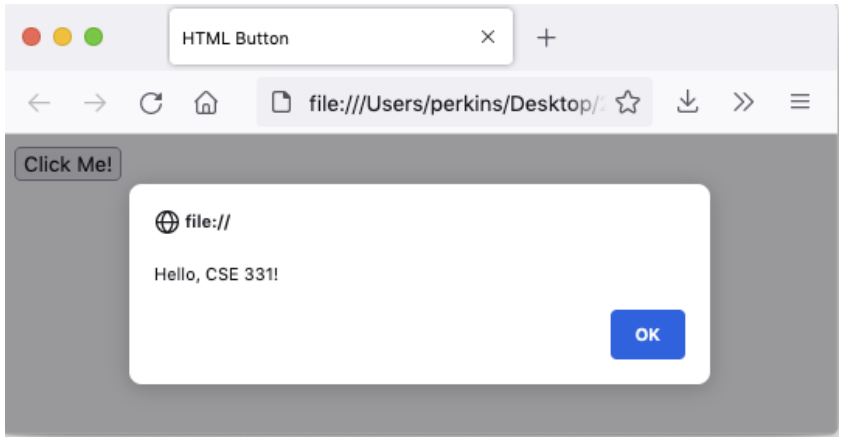



```
<html lang="en">
  <head>
    <title>331 Example Webpage</title>
  </head>
  <body>
    <h1>The Allen School</h1>
    <div>
      <p>
        The Allen School is a Computer Science school at
        UW. The best course in <br/> the Allen School is
        <a href="https://cs.uw.edu/331">CSE 331</a>.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

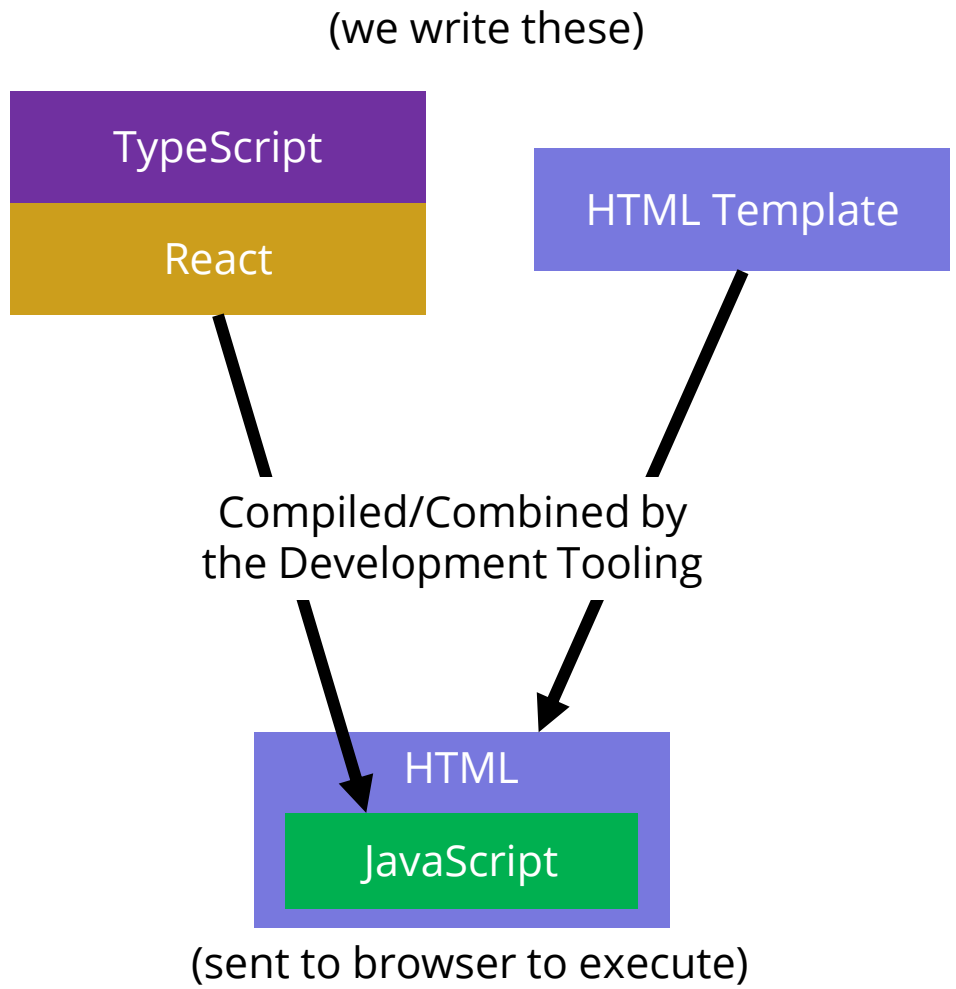
Demo 2

```
<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



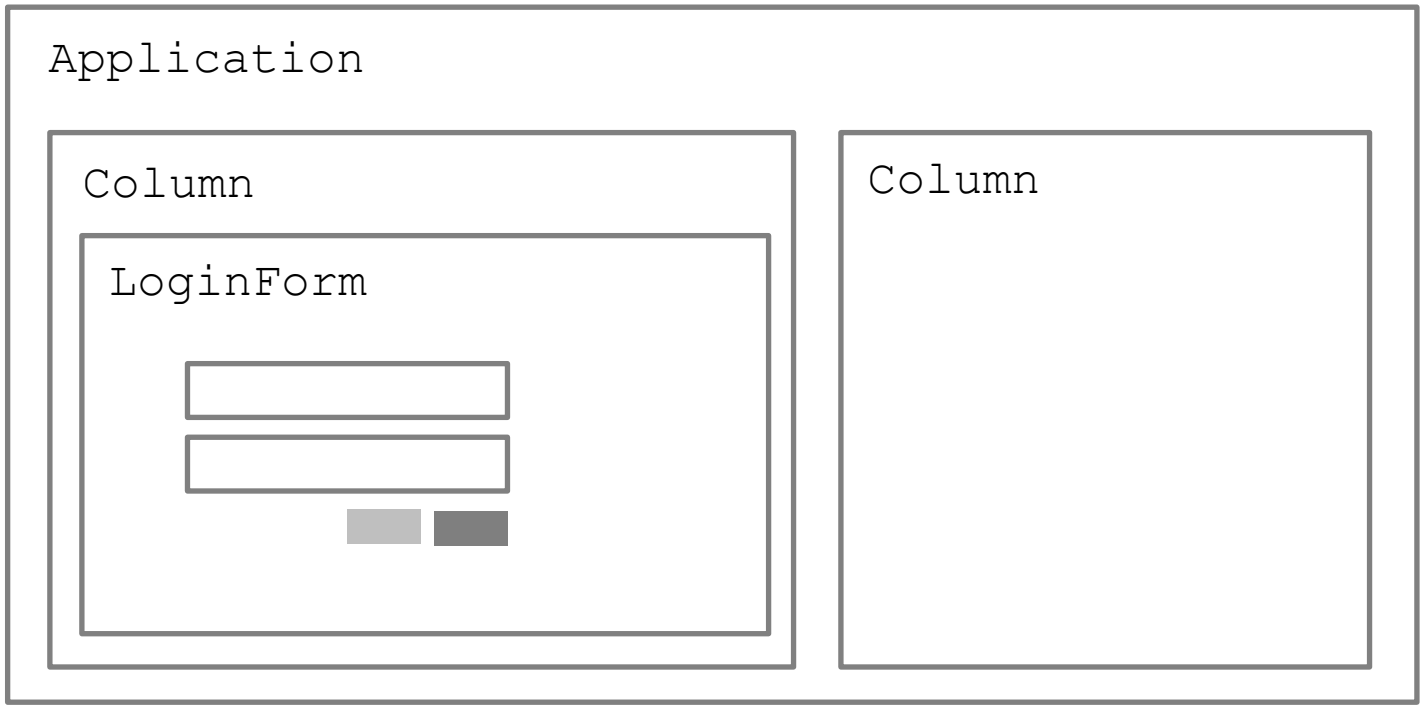
Example 1

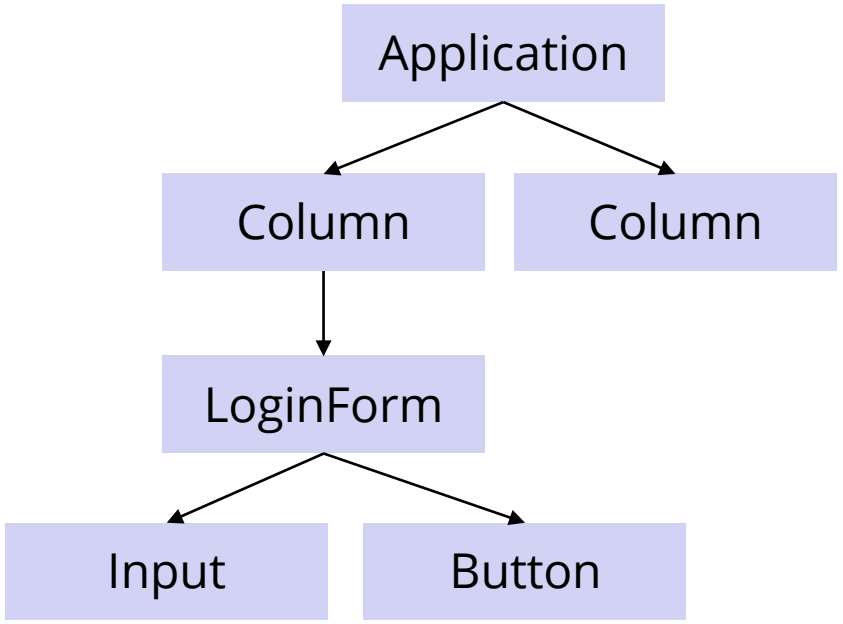
- The simplest source code to create a React website is these 3 files:
  - `index.html`
    - A very small amount of "necessary" HTML
    - Most of the actual web content will be generated by the TS/React code
  - `index.tsx`
    - Starting point of code – runs when the page loads
    - Starts React
  - `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

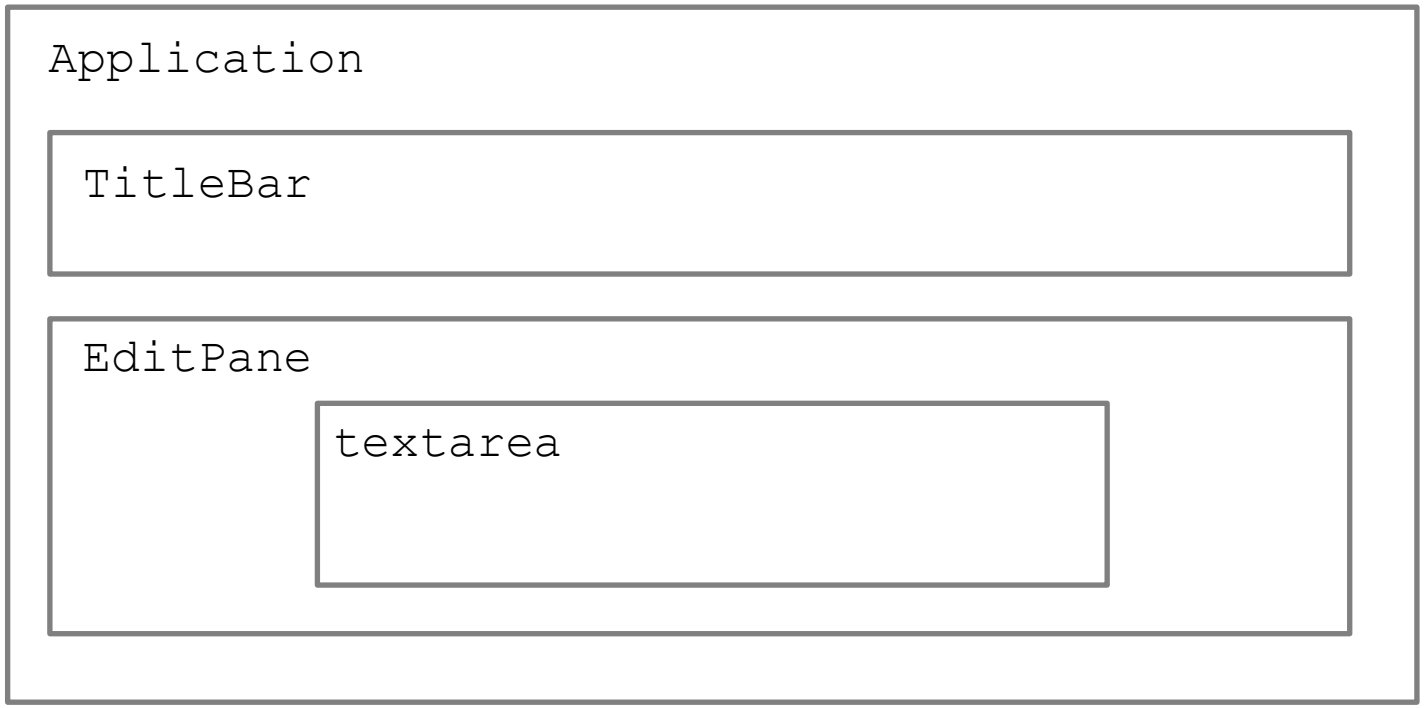
Components

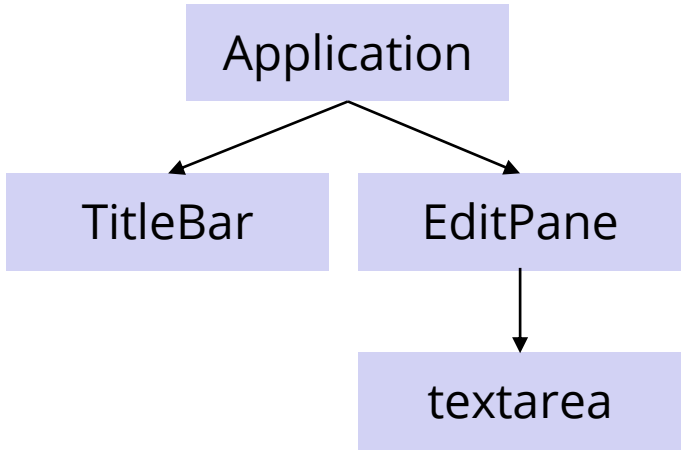
- We will have many components
  - e.g. Application, Column, LoginForm, Input, Button





Components


Application
TitleBar
EditPane
textarea

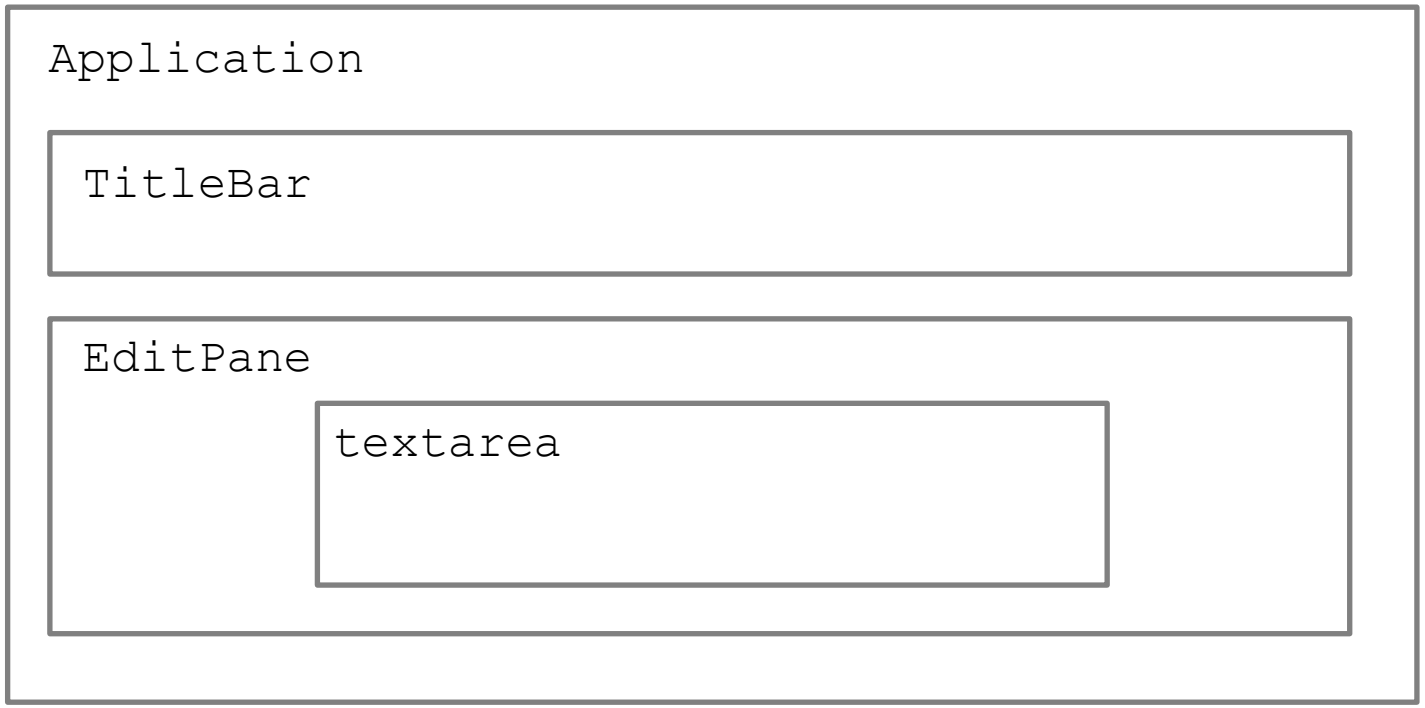

Application
TitleBar
EditPane
textarea

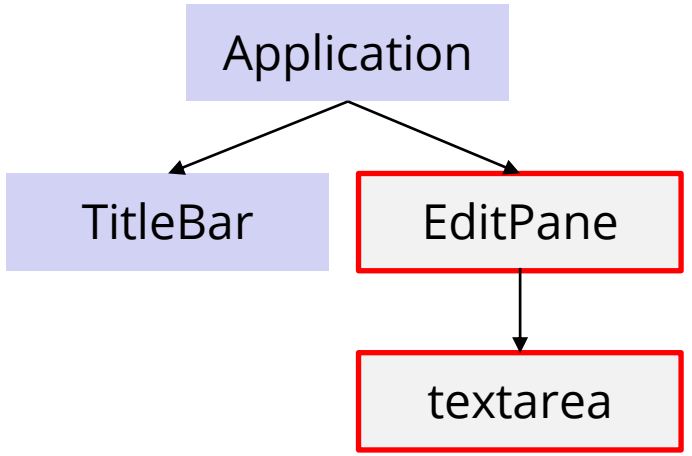
Example 3

register-react2/...

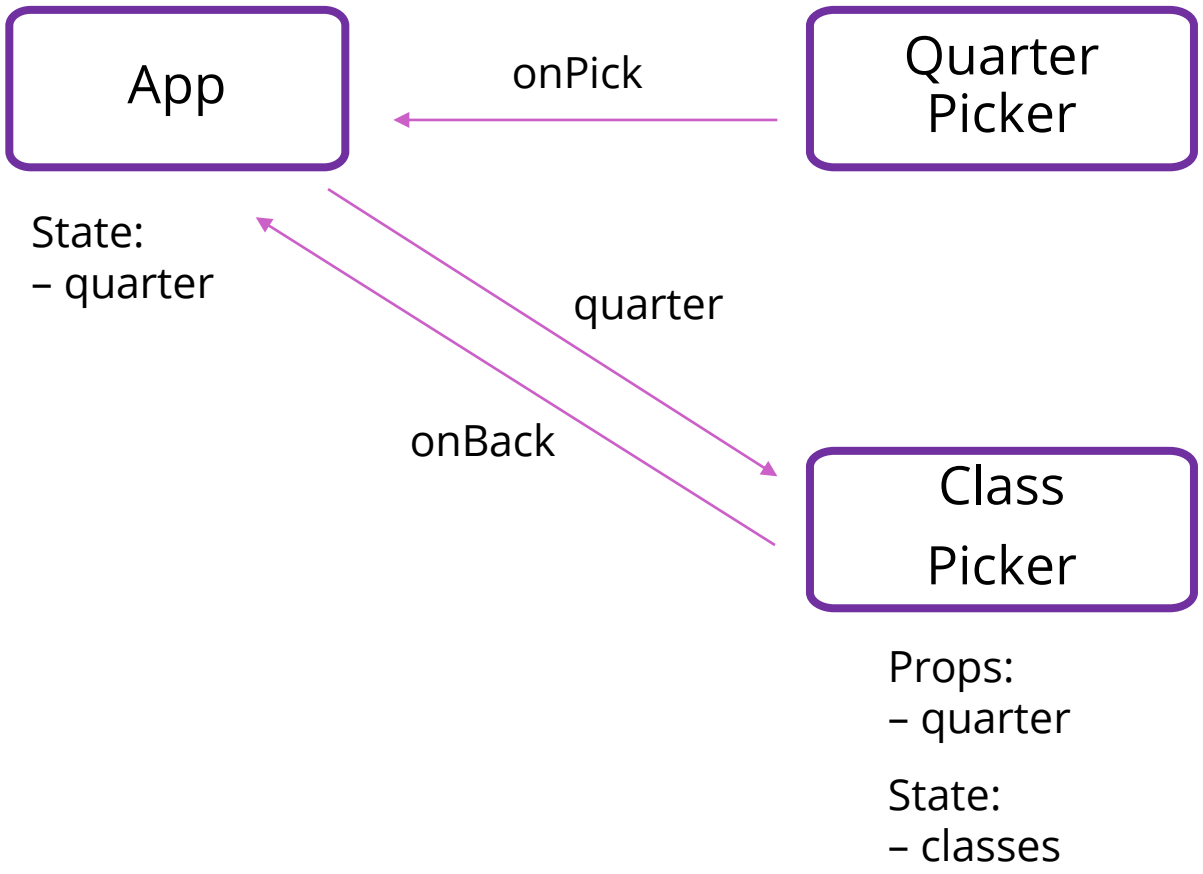
Passing Data from Child -> Parent

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





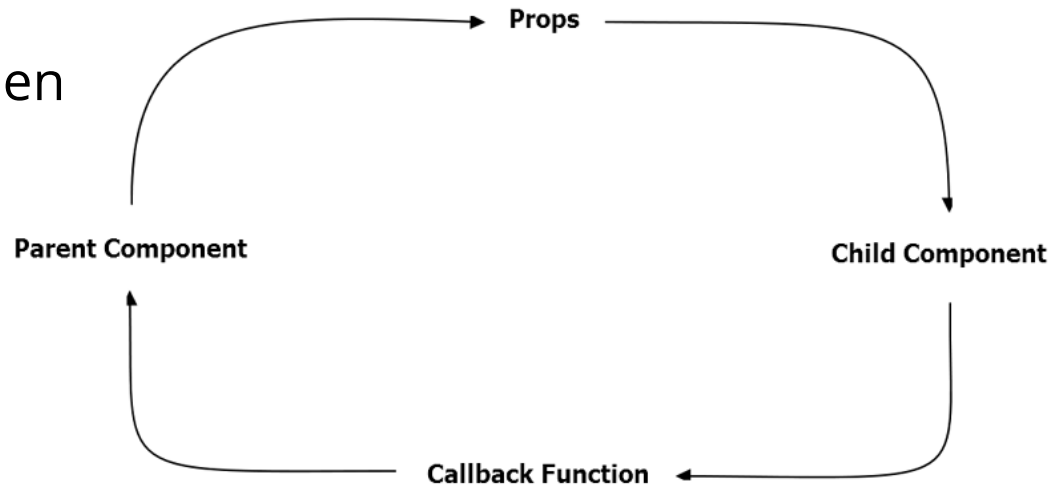
Structure of Example React App



Passing Around Information

- React terminology uses the term passing in (instead of registering) a callback function when we supply a function as a prop to a child component.

- We can propagate information upwards from child component.
  - Parent passes down a callback function from a parent component as a prop.
  - When called, the callback function can then update the fields (state) of the parent component from the child component.



Source: www.dotnettricks.com

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

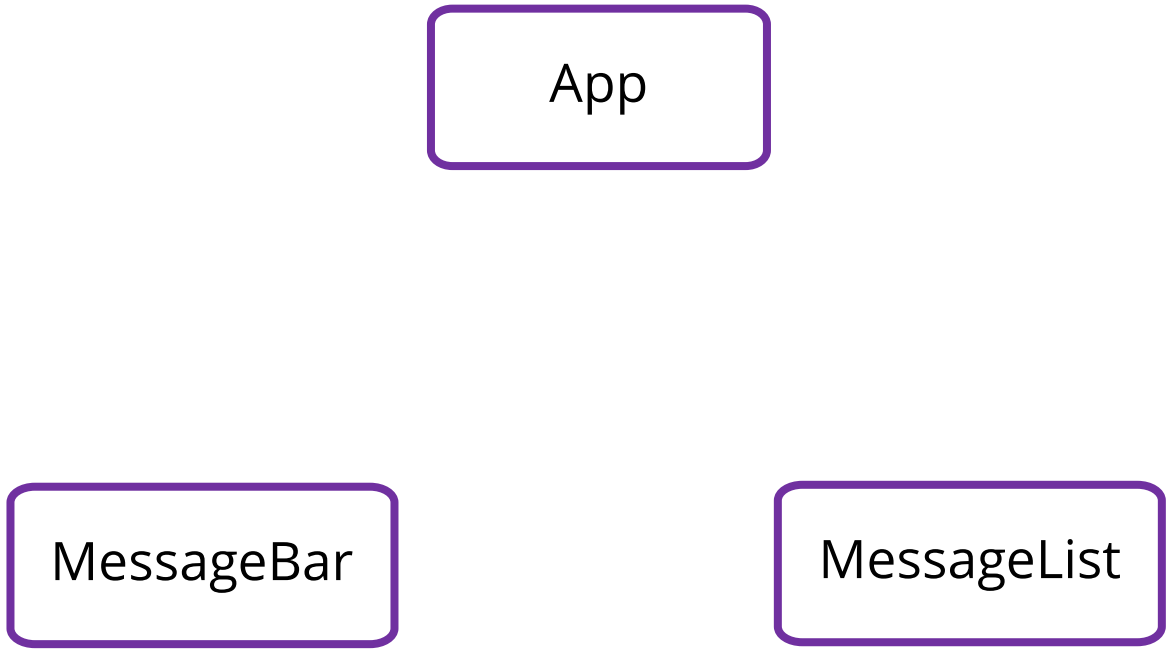
React Gotchas

- `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!

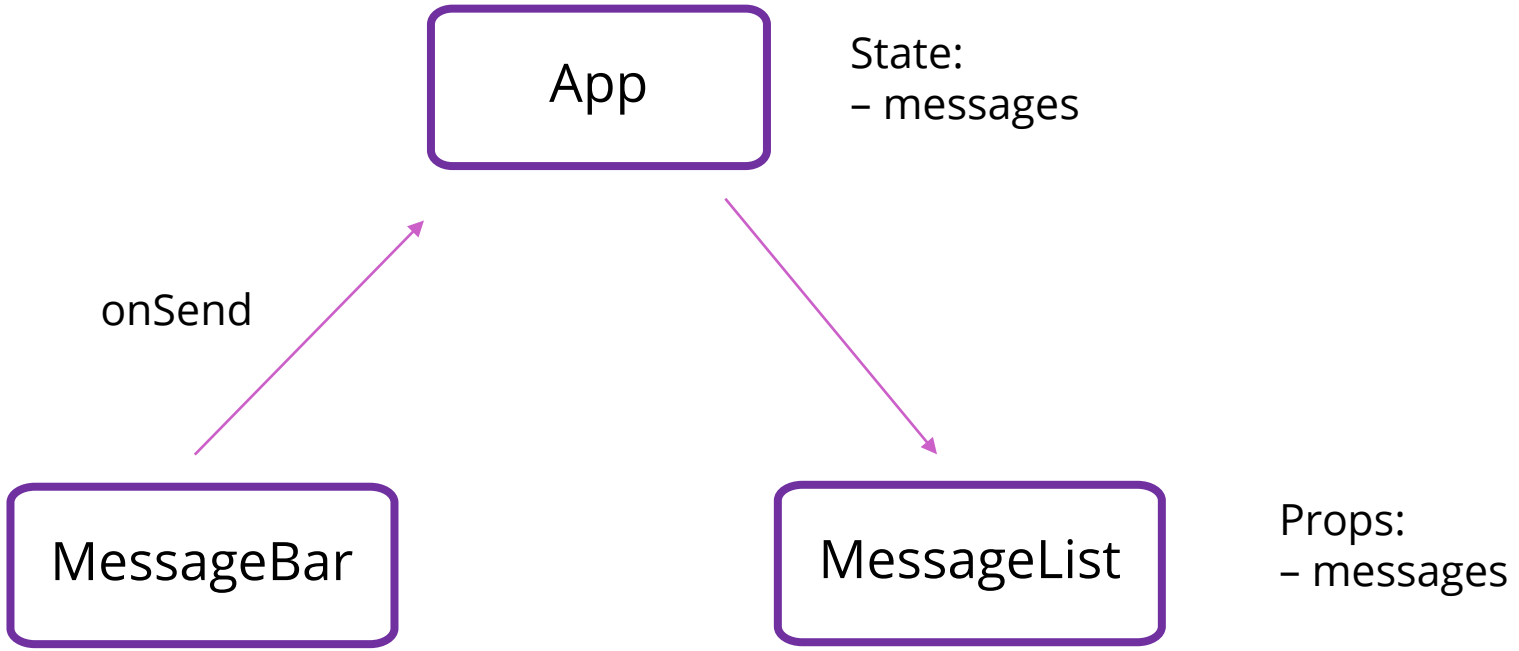
Example 4

messaging/...

How should data flow?



Data Flow


App
State:
– messages
onSend
MessageBar
MessageList
Props:
– messages

Before next class…

1. Start HW7 if you haven't already
   - Will need to apply generics
   - Useful for implementing Dijkstra's algorithm on a **`Graph<Double>`**