React Forms

- To code forms in React, you can use two techniques to create a component that "wraps" the HTML form:
    - **Controlled component** - the component holds the state as user updates the input
    - **Uncontrolled component** - the HTML DOM holds the state as user updates the input
- The React documentation recommends using **controlled components** as we demonstrate in the following example

test-react-form.html

In either case, since the component needs to maintain state, you cannot use a stateless function to define the component as we have done in earlier examples. It's easist to use ES6 classes instead.

It's typical in React for form submission to trigger an Ajax request, perhaps using axios to issue a GET, POST, UPDATE or DELETE. However, the example on this page just writes to the JavaScript console.

Using uncontrolled forms is perhaps simpler if you are porting existing non-React forms to React. For details, see:

https://facebook.github.io/react/docs/uncontrolled-components.html </p>

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

```
1    <!DOCTYPE html>
2    <html>
3      <head>
4        <meta charset="ISO-8859-1">
5        <title>Test Form</title>
6        <script src="https://unpkg.com/react@15.4.2/dist/react.js">
7        </script>
8        <script src="https://unpkg.com/react-dom@15.4.2/dist/react-dom.js">
9        </script>
10       <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.29/browser.
11       </script>
12       <script src="https://unpkg.com/axios/dist/axios.min.js">
13       </script>
14     </head>
15     <body>
16        <div id="target"></div>
17
18        <script type="text/babel">
19
20           /* use a ES6 class to define the component */
21           class StudentForm extends React.Component
22           {
23              constructor(props)
24              {
25                  super(props);
26
27                  /* initialize component's state */
28                  this.state =
29                  {
30                    name:'',
31                    gpa:'',
32                    id:''
33                  }
34
35                  /* bind the component's "this" so that within the
36                     functions, "this" refers to the component, not
37                     the function
38                   */
39                  this.handleSubmit =
40                     this.handleSubmit.bind(this);
41                  this.handleInputChange =
42                     this.handleInputChange.bind(this);
43              }
44
45              /* sync component data with the changes in the DOM */
```

```
46              handleInputChange(event)
47              {
48                 console.log("In handleInputChange: " + event);
49
50                 const target = event.target;
51                 const value = target.value;
52                 const name = target.name;
53
54                 this.setState(
55                 {
56                   [name]: value
57                 });
58              }
59
60              /* respond to submit */
61              handleSubmit(event)
62              {
63                  console.log("Submit: " + this.state.name +
64                        ", " + this.state.gpa +
65                        ", " + this.state.id );
66
67                  /* prevent actual form submission to server */
68                  event.preventDefault();
69              }
70
71              /* render the component */
72              render()
73              {
74                  return (
75                      <form onSubmit={this.handleSubmit}>
76                        <label>
77                          Name:
78                          <input
79                            name="name"
80                            type="text"
81                            value={this.state.name}
82                            onChange={this.handleInputChange} />
83                        </label>
84                        <br />
85                        <label>
86                          Student ID:
87                          <input
88                            name="id"
89                            type="text"
90                            value={this.state.id}
```

```
91                          onChange={this.handleInputChange} />
92                      </label>
93                      <br />
94                      <label>
95                        GPA:
96                        <input
97                          name="gpa"
98                          type="text"
99                          value={this.state.gpa}
100                         onChange={this.handleInputChange} />
101                     </label>
102                     <br />
103                     <input type="submit" value="Submit" />
104                   </form>
105                );
106           }
107         };
108
109         ReactDOM.render(
110            <StudentForm />,
111            document.getElementById('target'));
112
113      </script>
114   </body>
115 </html>
```