

## reacthello.html (Page 1 of 1)

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Hello</title>
5:   </head>
6:
7:   <body>
8:
9:     <div id="root"></div>
10:
11:     <script src=
12:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
.min.js">
13:     </script>
14:
15:     <script src=
16:       "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
oduction.min.js">
17:     </script>
18:
19:     <script src=
20:       "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
.js">
21:     </script>
22:
23:     <script type="text/babel">
24:
25:       'use strict';
26:
27:       //-----
28:
29:       function App() {
30:         return (
31:           <div>
32:             Hello, world.
33:           </div>
34:         );
35:       }
36:
37:       //-----
38:
39:       let domRoot = document.getElementById('root');
40:       let reactRoot = ReactDOM.createRoot(domRoot);
41:       reactRoot.render(
42:         <React.StrictMode>
43:           <App />
44:         </React.StrictMode>
45:       );
46:
47:     </script>
48:   </body>
49: </html>
50:
```

## blank (Page 1 of 1)

1: This page is intentionally blank.

## reactechol.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Echo</title>
5:   </head>
6:   <body>
7:     <div id="root"></div>
8:
9:     <script src=
10:      "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
.min.js">
11:     </script>
12:
13:     <script src=
14:      "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
oduction.min.js">
15:     </script>
16:
17:     <script src=
18:      "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
.js">
19:     </script>
20:
21:     <script type="text/babel">
22:
23:       'use strict';
24:
25:       //-----
26:
27:       function App() {
28:         const [enteredText, setEnteredText] = React.useState('');
29:
30:         return (
31:           <div>
32:             <h1>Echo</h1>
33:             Enter text to be echoed:&nbsp;   
34:             <input
35:               type='text'
36:               onInput={ function (event) {
37:                 setEnteredText(event.target.value);
38:               }}
39:               autoFocus
40:             />
41:             <hr />
42:             You entered: {enteredText}
43:           </div>
44:         );
45:       }
46:
47:       //-----
48:
49:       let domRoot = document.getElementById('root');
50:       let reactRoot = ReactDOM.createRoot(domRoot);
51:       reactRoot.render(
52:         <React.StrictMode>
53:         <App />
54:         </React.StrictMode>
55:       );
56:
57:     </script>
58:   </body>
59: </html>

```

## arrow.js (Page 1 of 1)

```

1: //-----
2: // arrow.js
3: // Author: Bob Dondero
4: //-----
5:
6: 'use strict';
7:
8: function square1(i) {return i * i;}
9: let square2 = function(i) {return i * i;};
10: let square3 = (i) => {return i * i;};
11: let square4 = (i) => i * i;
12: let square5 = i => i * i;
13:
14: function prod1(i, j) {return i * j;}
15: let prod2 = function(i, j) {return i * j;};
16: let prod3 = (i, j) => {return i * j;};
17: let prod4 = (i, j) => i * j;
18:
19: function greet1() {process.stdout.write('hi\n');}
20: let greet2 = function() {process.stdout.write('hi\n');};
21: let greet3 = () => {process.stdout.write('hi\n');};
22: let greet4 = () => process.stdout.write('hi\n');
23:
24: function main() {
25:   process.stdout.write(String(square1(5)) + '\n');
26:   process.stdout.write(String(square2(5)) + '\n');
27:   process.stdout.write(String(square3(5)) + '\n');
28:   process.stdout.write(String(square4(5)) + '\n');
29:   process.stdout.write(String(square5(5)) + '\n');
30:
31:   process.stdout.write(String(prod1(5, 6)) + '\n');
32:   process.stdout.write(String(prod2(5, 6)) + '\n');
33:   process.stdout.write(String(prod3(5, 6)) + '\n');
34:   process.stdout.write(String(prod4(5, 6)) + '\n');
35:
36:   greet1();
37:   greet2();
38:   greet3();
39:   greet4();
40: }
41:
42: if (require.main === module)
43:   main();

```

## reactecho2.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Echo</title>
5:   </head>
6:   <body>
7:     <div id="root"></div>
8:
9:     <script src=
10:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
11: .min.js">
12:   </script>
13:
14:   <script src=
15:     "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
16: oduction.min.js">
17:   </script>
18:
19:   <script src=
20:     "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
21: .js">
22:   </script>
23:   <script type="text/babel">
24:
25:     'use strict';
26:
27:     //-----
28:
29:     const App = () => {
30:       const [enteredText, setEnteredText] = React.useState('');
31:
32:       return (
33:         <div>
34:           <h1>Echo</h1>
35:           Enter text to be echoed:&nbsp;   
36:           <input
37:             type='text'
38:             onInput={ (event) => {
39:               setEnteredText(event.target.value);
40:             }}
41:             autoFocus
42:           />
43:           <hr />
44:           You entered: {enteredText}
45:         </div>
46:       );
47:     }
48:
49:     //-----
50:
51:     let domRoot = document.getElementById('root');
52:     let reactRoot = ReactDOM.createRoot(domRoot);
53:     reactRoot.render(
54:       <React.StrictMode>
55:         <App />
56:       </React.StrictMode>
57:     );
58:
59:   </script>
60: </body>
61: </html>

```

## reactecho3.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Echo</title>
5:   </head>
6:   <body>
7:     <div id="root"></div>
8:
9:     <script src=
10:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
11: .min.js">
12:   </script>
13:
14:   <script src=
15:     "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
16: oduction.min.js">
17:   </script>
18:
19:   <script src=
20:     "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
21: .js">
22:   </script>
23:   <script type="text/babel">
24:
25:     'use strict';
26:
27:     //-----
28:
29:     function App() {
30:       const [enteredText, setEnteredText] = React.useState('');
31:
32:       return (
33:         <div>
34:           <h1>Echo</h1>
35:           Enter text to be echoed:&nbsp;   
36:           <input
37:             type='text'
38:             onInput={ (event) => {
39:               setEnteredText(event.target.value);
40:             }}
41:             autoFocus
42:           />
43:           <hr />
44:           You entered: {enteredText}
45:         </div>
46:       );
47:     }
48:
49:     //-----
50:
51:     let domRoot = document.getElementById('root');
52:     let reactRoot = ReactDOM.createRoot(domRoot);
53:     reactRoot.render(
54:       <React.StrictMode>
55:         <App />
56:       </React.StrictMode>
57:     );
58:
59:   </script>
60: </body>
61: </html>

```

## reactecho4.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Echo Props</title>
5:   </head>
6:   <body>
7:     <div id="root"></div>
8:
9:     <script src=
10:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
11: .min.js">
12:   </script>
13:
14:   <script src=
15:     "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
16: oduction.min.js">
17:   </script>
18:
19:   <script src=
20:     "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
21: .js">
22:   </script>
23:   <script type="text/babel">
24:
25:     'use strict';
26:
27:     //-----
28:
29:     function EchoInput(props) {
30:       return (
31:         <div>
32:           <h1>Echo Props</h1>
33:           Enter text to be echoed:&nbsp;   
34:           <input
35:             type='text'
36:             onInput={ (event) => {
37:               props.callback(event.target.value);
38:             }}
39:             autoFocus
40:           />
41:         </div>
42:       );
43:     }
44:
45:     //-----
46:
47:     function EchoOutput(props) {
48:       return (
49:         <div>
50:           {props.txt}
51:         </div>
52:       );
53:     }
54:
55:     //-----
56:
57:     function Echo() {
58:       const [enteredText, setEnteredText] = React.useState('');
59:
60:       return (
61:         <div>
62:           <EchoInput callback={setEnteredText} />

```

## reactecho4.html (Page 2 of 2)

```

63:           <hr />
64:           <EchoOutput txt={enteredText} />
65:         </div>
66:       );
67:     }
68:
69:     //-----
70:
71:     function App() {
72:       return (
73:         <Echo />
74:       );
75:     }
76:
77:     //-----
78:
79:     let domRoot = document.getElementById('root');
80:     let reactRoot = ReactDOM.createRoot(domRoot);
81:     reactRoot.render(
82:       <React.StrictMode>
83:         <App />
84:       </React.StrictMode>
85:     );
86:
87:   </script>
88: </body>
89: </html>
90:

```

## reactdatetime.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Echo</title>
5:   </head>
6:
7:   <body>
8:
9:     <div id="root"></div>
10:
11:     <script src=
12:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
.min.js">
13:     </script>
14:
15:     <script src=
16:       "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
oduction.min.js">
17:     </script>
18:
19:     <script src=
20:       "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
.js">
21:     </script>
22:
23:     <script type="text/babel">
24:
25:       'use strict';
26:
27:       //-----
28:
29:       function App() {
30:         const [datetime, setDatetime] = React.useState(new Date());
31:
32:         function updateDatetime() {
33:           window.setInterval(
34:             () => {setDatetime(new Date());},
35:             1000
36:           );
37:         }
38:
39:         React.useEffect(updateDatetime, []);
40:
41:         let hours = datetime.getHours();
42:         let ampm = (hours < 12) ? 'morning' : 'afternoon';
43:         return (
44:           <div>
45:             <h1>Good {ampm}</h1>
46:             The current date and time are<br/>
47:             {datetime.toLocaleString()}
48:           </div>
49:         );
50:       }
51:
52:       //-----
53:
54:       let domRoot = document.getElementById('root');
55:       let reactRoot = ReactDOM.createRoot(domRoot);
56:       reactRoot.render(
57:         <React.StrictMode>
58:           <App />
59:         </React.StrictMode>
60:       );
61:
62: </script>

```

## reactdatetime.html (Page 2 of 2)

```

63:   </body>
64: </html>
65:

```

## PennyReact1/index.html (Page 1 of 3)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <div id="root"></div>
8:
9:     <script src=
10:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
11: .min.js">
12:     </script>
13:
14:     <script src=
15:       "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
16: oduction.min.js">
17:     </script>
18:
19:     <script src=
20:       "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
21: .js">
22:     </script>
23:     <script type="text/babel">
24:
25:       'use strict';
26:
27:       //-----
28:
29:       function PennyHeader() {
30:         const [datetime, setDatetime] = React.useState(new Date());
31:
32:         function updateHeader() {
33:           window.setInterval(
34:             () => {setDatetime(new Date());},
35:             1000
36:           );
37:         }
38:
39:         React.useEffect(updateHeader, []);
40:
41:         let hours = datetime.getHours();
42:         let amp = (hours < 12) ? 'morning' : 'afternoon';
43:         return (
44:           <div>
45:             <hr />
46:             Good {amp} and welcome to Penny.com
47:             <hr />
48:           </div>
49:         );
50:       }
51:
52:       //-----
53:
54:       function PennySearch() {
55:         const [author, setAuthor] = React.useState('');
56:         const [books, setBooks] = React.useState([]);
57:
58:         function fetchBooks() {
59:           function handleResponse() {
60:             if (this.status !== 200) {
61:               alert('Error: Failed to fetch data from server');
62:             }

```

## PennyReact1/index.html (Page 2 of 3)

```

63:           }
64:           let books = JSON.parse(this.response);
65:           setBooks(books);
66:         }
67:
68:         function handleError() {
69:           alert('Error: Failed to fetch data from server');
70:         }
71:
72:         let encodedAuthor = encodeURIComponent(author);
73:         let url = '/searchresults?author=' + encodedAuthor;
74:         let request = new XMLHttpRequest();
75:         request.onload = handleResponse;
76:         request.onerror = handleError;
77:         request.open('GET', url);
78:         request.send();
79:         return () => {request.abort();}
80:       }
81:
82:       React.useEffect(fetchBooks, [author]);
83:
84:       let timer = null;
85:
86:       function debouncedSetAuthor(author) {
87:         clearTimeout(timer);
88:         timer = setTimeout(()=>{setAuthor(author);}, 500);
89:       }
90:
91:       return (
92:         <div>
93:           <h1>Author Search</h1>
94:           Please enter an author name:&nbsp;&nbsp;&nbsp;
95:           <input
96:             type='text'
97:             onInput={ (event) => {
98:               debouncedSetAuthor(event.target.value);
99:             }
100:             autoFocus
101:           />
102:           <hr />
103:           {books.map((book) => (
104:             <div key={book.isbn}>
105:               {book.isbn} &nbsp;&nbsp;&nbsp;
106:               <strong>{book.author}</strong> &nbsp;&nbsp;&nbsp;
107:               {book.title}
108:             <br />
109:             </div>
110:           ))}
111:         </div>
112:       );
113:     }
114:
115:     //-----
116:
117:     function PennyFooter() {
118:       const [datetime, setDatetime] = React.useState(new Date());
119:
120:       function updateFooter() {
121:         window.setInterval(
122:           () => {setDatetime(new Date());},
123:           1000
124:         );
125:       }
126:
127:       React.useEffect(updateFooter, []);

```

## PennyReact1/index.html (Page 3 of 3)

```
128:         return (
129:             <div>
130:                 <hr />
131:                 Date and time: {datetime.toLocaleString()}
132:                 <br />
133:                 Created by&nbsp;&nbsp;&nbsp;
134:                 <a href="https://www.cs.princeton.edu/~rdondero">
135:                 Bob Dondero</a>
136:             </div>
137:         );
138:     }
139: }
140:
141: //-----
142:
143: function App() {
144:     return (
145:         <div>
146:             <PennyHeader />
147:             <PennySearch />
148:             <PennyFooter />
149:         </div>
150:     );
151: }
152:
153: //-----
154:
155: let domRoot = document.getElementById('root');
156: let reactRoot = ReactDOM.createRoot(domRoot);
157: reactRoot.render(
158:     <React.StrictMode>
159:     <App />
160: </React.StrictMode>
161: );
162:
163: </script>
164: </body>
165: </html>
166:
```

## blank (Page 1 of 1)

1: This page is intentionally blank.

## PennyReact2/index.html (Page 1 of 3)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <div id="root"></div>
8:
9:     <script src=
10:       "https://cdn.jsdelivr.net/npm/react@18.3.1/umd/react.production
11: .min.js">
12:   </script>
13:
14:   <script src=
15:     "https://cdn.jsdelivr.net/npm/react-dom@18.3.1/umd/react-dom.pr
16: oduction.min.js">
17:   </script>
18:
19:   <script src=
20:     "https://cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min
21: .js">
22:   </script>
23:
24:   <script type="text/babel">
25:     'use strict';
26:
27:     //-----
28:
29:     function PennyHeader() {
30:       const [datetime, setDatetime] = React.useState(new Date());
31:
32:       function updateHeader() {
33:         window.setInterval(
34:           () => {setDatetime(new Date());},
35:           1000
36:         );
37:       }
38:
39:       React.useEffect(updateHeader, []);
40:
41:       let hours = datetime.getHours();
42:       let ampm = (hours < 12) ? 'morning' : 'afternoon';
43:       return (
44:         <div>
45:           <hr />
46:           Good {ampm} and welcome to Penny.com
47:           <hr />
48:         </div>
49:       );
50:     }
51:
52:     //-----
53:
54:     function PennyInput(props) {
55:       return (
56:         <div>
57:           <h1>Author Search</h1>
58:           Please enter an author name:&nbsp;&nbsp;&nbsp;
59:           <input
60:             type='text'
61:             onInput={ (event) => {
62:               props.callback(event.target.value);

```

## PennyReact2/index.html (Page 2 of 3)

```

63:         }}
64:         autoFocus
65:       />
66:     <hr />
67:   </div>
68: );
69: }
70:
71: //-----
72:
73: function PennyOutput(props) {
74:   return (
75:     <div>
76:       {props.books.map((book) => (
77:         <div key={book.isbn}>
78:           {book.isbn}&nbsp;&nbsp;&nbsp;
79:           <strong>{book.author}</strong>&nbsp;&nbsp;&nbsp;
80:           {book.title}
81:           <br />
82:         </div>
83:       ))}
84:     </div>
85:   );
86: }
87:
88: //-----
89:
90: function PennySearch() {
91:   const [author, setAuthor] = React.useState('');
92:   const [books, setBooks] = React.useState([]);
93:
94:   function fetchBooks() {
95:     function handleResponse() {
96:       if (this.status !== 200) {
97:         alert('Error: Failed to fetch data from server');
98:         return;
99:       }
100:      let books = JSON.parse(this.response);
101:      setBooks(books);
102:    }
103:
104:    function handleError() {
105:      alert('Error: Failed to fetch data from server');
106:    }
107:
108:    let encodedAuthor = encodeURIComponent(author);
109:    let url = '/searchresults?author=' + encodedAuthor;
110:    let request = new XMLHttpRequest();
111:    request.onload = handleResponse;
112:    request.onerror = handleError;
113:    request.open('GET', url);
114:    request.send();
115:    return () => {request.abort();}
116:  }
117:
118:  React.useEffect(fetchBooks, [author]);
119:
120:  let timer = null;
121:
122:  function debouncedSetAuthor(author) {
123:    clearTimeout(timer);
124:    timer = setTimeout(()=>{setAuthor(author);}, 500);
125:  }
126:
127:  return (

```



## PennyReact2/index.html (Page 3 of 3)

```

128:         <div>
129:             <PennyInput callback={debouncedSetAuthor} />
130:             <PennyOutput books={books} />
131:         </div>
132:     );
133: }
134:
135: //-----
136:
137: function PennyFooter() {
138:     const [datetime, setDatetime] = React.useState(new Date());
139:
140:     function updateFooter() {
141:         window.setInterval(
142:             () => {setDatetime(new Date());},
143:             1000
144:         );
145:     }
146:
147:     React.useEffect(updateFooter, []);
148:
149:     return (
150:         <div>
151:             <hr />
152:             Date and time: {datetime.toLocaleString()}
153:             <br />
154:             Created by&nbsp;
155:             <a href="https://www.cs.princeton.edu/~rdondero">
156:                 Bob Dondero</a>
157:         </div>
158:     );
159: }
160:
161: //-----
162:
163: function App() {
164:     return (
165:         <div>
166:             <PennyHeader />
167:             <PennySearch />
168:             <PennyFooter />
169:         </div>
170:     );
171: }
172:
173: //-----
174:
175: let domRoot = document.getElementById('root');
176: let reactRoot = ReactDOM.createRoot(domRoot);
177: reactRoot.render(
178:     <React.StrictMode>
179:         <App />
180:     </React.StrictMode>
181: );
182:
183: </script>
184: </body>
185: </html>
186:

```

## blank (Page 1 of 1)

1: This page is intentionally blank.

## arrow1.js (Page 1 of 1)

```
1: //-----
2: // arrow1.js
3: // Author: Bob Dondero
4: //-----
5:
6: 'use strict';
7:
8: class Car {
9:
10:   constructor(color) {
11:     this._color = color;
12:   }
13:
14:   writeColor() {
15:     // Error: this is undefined.
16:     setTimeout(
17:       function () { process.stdout.write(this._color + '\n'); },
18:       2000
19:     );
20:   }
21: }
22:
23: function main() {
24:   let blueCar = new Car('blue');
25:   blueCar.writeColor();
26: }
27:
28: if (require.main === module)
29:   main();
```

## arrow2.js (Page 1 of 1)

```
1: //-----
2: // arrow2.js
3: // Author: Bob Dondero
4: //-----
5:
6: 'use strict';
7:
8: class Car {
9:
10:   constructor(color) {
11:     this._color = color;
12:   }
13:
14:   writeColor() {
15:     let self = this;
16:     setTimeout(
17:       function () { process.stdout.write(self._color + '\n'); },
18:       2000
19:     );
20:   }
21: }
22:
23: function main()
24: {
25:   let blueCar = new Car('blue');
26:   blueCar.writeColor();
27: }
28:
29: if (require.main === module)
30:   main();
```

## arrow3.js (Page 1 of 1)

```
1: //-----  
2: // arrow3.js  
3: // Author: Bob Dondero  
4: //-----  
5:  
6: 'use strict';  
7:  
8: class Car {  
9:  
10:   constructor(color) {  
11:     this._color = color;  
12:   }  
13:  
14:   writeColor() {  
15:     setTimeout(  
16:       () => { process.stdout.write(this._color + '\n'); },  
17:       2000  
18:     );  
19:   }  
20: }  
21:  
22: function main() {  
23:   let blueCar = new Car('blue');  
24:   blueCar.writeColor();  
25: }  
26:  
27: if (require.main === module)  
28:   main();
```