

**noreacthello.html (Page 1 of 1)**

```
1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Hello</title>
5:   </head>
6:
7:   <body>
8:     Hello, world.
9:   </body>
10: </html>
```

**blank (Page 1 of 1)**

```
1: This page is intentionally blank.
```

## 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:
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 [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={ function (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>

```

## blank (Page 1 of 1)

1: This page is intentionally blank.

## arrow.js (Page 1 of 1)

```
1: //-----
2: // arrow.js
3: // Author: Bob Dundero
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();
```

## blank (Page 1 of 1)

```
1: This page is intentionally blank.
```

## reactecho2.html (Page 1 of 1)

```

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:       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>

```

## blank (Page 1 of 1)

1: This page is intentionally blank.

## reactecho3.html (Page 1 of 1)

```

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 [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>

```

## blank (Page 1 of 1)

1: This page is intentionally blank.

## reactecho4.html (Page 1 of 2)

```

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 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:

```



## noreactdatetime.html (Page 1 of 1)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>DateTime</title>
5:   </head>
6:
7:   <body>
8:     <h1>Good <span id="ampmSpan"></span></h1>
9:     The current date and time are<br>
10:    <span id="datetimeSpan"></span>
11:    <br/>
12:
13:    <script>
14:      'use strict';
15:
16:      function getAmPm() {
17:        let dateTime = new Date();
18:        let hours = dateTime.getHours();
19:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
20:        let ampMspan = document.getElementById('ampmSpan');
21:        ampMspan.innerHTML = amPm;
22:      }
23:
24:      function getDateTime() {
25:        let dateTime = new Date();
26:        let datetimeSpan =
27:          document.getElementById('datetimeSpan');
28:        datetimeSpan.innerHTML = dateTime.toLocaleString();
29:      }
30:
31:      function setupAmpm() {
32:        getAmPm();
33:        let apInterval = window.setInterval(getAmPm, 1000);
34:        window.addEventListener('beforeunload',
35:          function(e) {window.clearInterval(apInterval);})
36:      }
37:
38:      function setupDatetime() {
39:        getDateTime();
40:        let dtInterval = window.setInterval(getDateTime, 1000);
41:        window.addEventListener('beforeunload',
42:          function(e) {window.clearInterval(dtInterval);})
43:      }
44:
45:      function setup() {
46:        setupAmpm();
47:        setupDatetime();
48:      }
49:
50:      document.addEventListener('DOMContentLoaded', setup);
51:    </script>
52:
53:  </body>
54: </html>

```

## blank (Page 1 of 1)

```

1: This page is intentionally blank.

```

## reactdatetime.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>DateTime</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:           let dtInterval = window.setInterval(
34:             () => {setDatetime(new Date());},
35:             1000
36:           );
37:           return () => {window.clearInterval(dtInterval);};
38:         }
39:
40:         React.useEffect(updateDatetime, []);
41:
42:         let hours = datetime.getHours();
43:         let ampm = (hours < 12) ? 'morning' : 'afternoon';
44:         return (
45:           <div>
46:             <h1>Good {ampm}</h1>
47:             The current date and time are<br/>
48:             {datetime.toLocaleString()}
49:           </div>
50:         );
51:       }
52:
53:       //-----
54:
55:       let domRoot = document.getElementById('root');
56:       let reactRoot = ReactDOM.createRoot(domRoot);
57:       reactRoot.render(
58:         <React.StrictMode>
59:           <App />
60:         </React.StrictMode>
61:       );
62:

```

## reactdatetime.html (Page 2 of 2)

```

63:     </script>
64:   </body>
65: </html>
66:

```

## 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:
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:           let apInterval = window.setInterval(
34:             () => {setDatetime(new Date());},
35:             1000
36:           );
37:           return () => {window.clearInterval(apInterval)};
38:         }
39:
40:         React.useEffect(updateHeader, []);
41:
42:         let hours = datetime.getHours();
43:         let ampM = (hours < 12) ? 'morning' : 'afternoon';
44:         return (
45:           <div>
46:             <hr />
47:             Good {ampM} and welcome to Penny.com
48:             <hr />
49:           </div>
50:         );
51:       }
52:
53:       //-----
54:
55:       function PennySearch() {
56:         const [books, setBooks] = React.useState([]);
57:
58:         let request = null;
59:
60:         function fetchBooks(author) {
61:           function handleResponse() {
62:             if (this.status !== 200) {

```

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

```

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

```

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

```
128:     }
129:
130:     React.useEffect(updateFooter, []);
131:
132:     return (
133:       <div>
134:         <hr />
135:         Date and time: {datetime.toLocaleString()}
136:         <br />
137:         Created by&nbsp;&nbsp;&nbsp;
138:         <a href="https://www.cs.princeton.edu/~rdondero">
139:           Bob Dondero</a>
140:       </div>
141:     );
142:   }
143:
144: //-----
145:
146:   function App() {
147:     return (
148:       <div>
149:         <PennyHeader />
150:         <PennySearch />
151:         <PennyFooter />
152:       </div>
153:     );
154:   }
155:
156: //-----
157:
158:   let domRoot = document.getElementById('root');
159:   let reactRoot = ReactDOM.createRoot(domRoot);
160:   reactRoot.render(
161:     <React.StrictMode>
162:       <App />
163:     </React.StrictMode>
164:   );
165:
166: </script>
167: </body>
168: </html>
169:
```

## 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:           let apInterval = window.setInterval(
34:             () => {setDatetime(new Date());},
35:             1000
36:           );
37:           return () => {window.clearInterval(apInterval);};
38:         }
39:
40:         React.useEffect(updateHeader, []);
41:
42:         let hours = datetime.getHours();
43:         let ampm = (hours < 12) ? 'morning' : 'afternoon';
44:         return (
45:           <div>
46:             <hr />
47:             Good {ampm} and welcome to Penny.com
48:             <hr />
49:           </div>
50:         );
51:       }
52:
53:       //-----
54:
55:       function PennyInput(props) {
56:         return (
57:           <div>
58:             <h1>Author Search</h1>
59:             Please enter an author name:&nbsp;&nbsp;&nbsp;
60:             <input
61:               type='text'
62:               onInput={ (event) => {

```

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

```

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

```

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

```

128:         return (
129:             <div>
130:                 <PennyInput callback={debouncedFetchBooks} />
131:                 <PennyOutput books={books} />
132:             </div>
133:         );
134:     }
135:
136:     //-----
137:
138:     function PennyFooter() {
139:         const [datetime, setDatetime] = React.useState(new Date());
140:
141:         function updateFooter() {
142:             let dtInterval = window.setInterval(
143:                 () => {setDatetime(new Date());},
144:                 1000
145:             );
146:             return () => {window.clearInterval(dtInterval)};
147:         }
148:
149:         React.useEffect(updateFooter, []);
150:
151:         return (
152:             <div>
153:                 <hr />
154:                 Date and time: {datetime.toLocaleString()}
155:                 <br />
156:                 Created by &nbsp;&nbsp;
157:                 <a href="https://www.cs.princeton.edu/~rdondero">
158:                     Bob Dondero</a>
159:             </div>
160:         );
161:     }
162:
163:     //-----
164:
165:     function App() {
166:         return (
167:             <div>
168:                 <PennyHeader />
169:                 <PennySearch />
170:                 <PennyFooter />
171:             </div>
172:         );
173:     }
174:
175:     //-----
176:
177:     let domRoot = document.getElementById('root');
178:     let reactRoot = ReactDOM.createRoot(domRoot);
179:     reactRoot.render(
180:         <React.StrictMode>
181:             <App />
182:         </React.StrictMode>
183:     );
184:
185: </script>
186: </body>
187: </html>
188:

```

## 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();
```