

```
1: <!--
2:
3: ajax1.html
4:
5: Gets stock quote from quotel.php via Ajax, displaying result with alert().
6:
7: David J. Malan
8: Dan Armendariz
9: Computer Science E-75
10: Harvard Extension School
11:
12: -->
13:
14: <!DOCTYPE html PUBLIC
15:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
16:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
17:
18: <html xmlns="http://www.w3.org/1999/xhtml">
19:   <head>
20:     <script type="text/javascript">
21:       // <![CDATA[
22:
23:         // an XMLHttpRequest
24:         var xhr = null;
25:
26:         /*
27:          * void
28:          * quote()
29:          *
30:          * Gets a quote.
31:          */
32:         function quote()
33:         {
34:           // instantiate XMLHttpRequest object
35:           try
36:           {
37:             xhr = new XMLHttpRequest();
38:           }
39:           catch (e)
40:           {
41:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
42:           }
43:
44:           // handle old browsers
45:           if (xhr == null)
46:           {
47:             alert("Ajax not supported by your browser!");
```

```
48:             return;
49:           }
50:
51:           // construct URL
52:           var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
53:
54:           // get quote
55:           xhr.onreadystatechange = handler;
56:           xhr.open("GET", url, true);
57:           xhr.send(null);
58:         }
59:
60:         /*
61:          * void
62:          * handler()
63:          *
64:          * Handles the Ajax response.
65:          */
66:         function handler()
67:         {
68:           // only handle loaded requests
69:           if (xhr.readyState == 4)
70:           {
71:             if (xhr.status == 200)
72:               alert(xhr.responseText);
73:             else
74:               alert("Error with Ajax call!");
75:           }
76:         }
77:
78:       // ]]>
79:     </script>
80:   <title></title>
81: </head>
82: <body>
83:   <form action="quotel.php" method="get" onsubmit="quote(); return false;">
84:     Symbol: <input id="symbol" name="symbol" type="text" />
85:     <br /><br />
86:     <input type="submit" value="Get Quote" />
87:   </form>
88: </body>
89: </html>
```

```
1: <!--
2:
3: ajax10.html
4:
5: Gets stock quote from quote6.php via Ajax, inserting result into DOM.
6: Also displays returned JSON in a textarea.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20: <head>
21: <script type="text/javascript">
22:     // <![CDATA[
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:             // instantiate XMLHttpRequest object
36:             try
37:             {
38:                 xhr = new XMLHttpRequest();
39:             }
40:             catch (e)
41:             {
42:                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:             }
44:
45:             // handle old browsers
46:             if (xhr == null)
47:             {
```

```
48:             alert("Ajax not supported by your browser!");
49:             return;
50:         }
51:
52:         // get symbol
53:         var symbol = document.getElementById("symbol").value;
54:
55:         // construct URL
56:         var url = "quote6.php?symbol=" + symbol;
57:
58:         // get quote
59:         xhr.onreadystatechange =
60:         function()
61:         {
62:             // only handle loaded requests
63:             if (xhr.readyState == 4)
64:             {
65:                 if (xhr.status == 200)
66:                 {
67:                     // evaluate JSON
68:                     var quote = eval("(" + xhr.responseText + ")");
69:
70:                     // show JSON in textarea
71:                     document.getElementById("code").value = xhr.responseText;
72:
73:                     // insert quote into DOM
74:                     var div = document.createElement("div");
75:                     var text = document.createTextNode(symbol + ": " + quote.price);
76:                     div.appendChild(text);
77:                     document.getElementById("quotes").appendChild(div);
78:                 }
79:                 else
80:                 alert("Error with Ajax call!");
81:             }
82:         }
83:         xhr.open("GET", url, true);
84:         xhr.send(null);
85:     }
86:
87: // ]]>
88: </script>
89: <title></title>
90: </head>
91: <body>
92: <form onsubmit="quote(); return false;">
93:     symbol: <input id="symbol" type="text" />
94:     <br /><br />
```

```
95:     <input type="submit" value="Get Quote" />
96:   </form>
97:   <br /><br />
98:   <div id="quotes"></div>
99:   <br /><br />
100:  <textarea cols="80" id="code" rows="16"></textarea>
101: </body>
102: </html>
```

```
1: <!--
2:
3: ajax11.html
4:
5: Gets stock quote from quote7.php via Ajax, inserting result into DOM.
6: Also displays returned JSON in a textarea.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20:   <head>
21:     <script type="text/javascript">
22:       // <![CDATA[
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:           // instantiate XMLHttpRequest object
36:           try
37:           {
38:             xhr = new XMLHttpRequest();
39:           }
40:           catch (e)
41:           {
42:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:           }
44:
45:           // handle old browsers
46:           if (xhr == null)
47:           {
```

```
48:         alert("Ajax not supported by your browser!");
49:         return;
50:     }
51:
52:     // get symbol
53:     var symbol = document.getElementById("symbol").value;
54:
55:     // construct URL
56:     var url = "quote7.php?symbol=" + symbol;
57:
58:     // get quote
59:     xhr.onreadystatechange =
60:     function()
61:     {
62:         // only handle loaded requests
63:         if (xhr.readyState == 4)
64:         {
65:             if (xhr.status == 200)
66:             {
67:                 // evaluate JSON
68:                 var quote = eval("(" + xhr.responseText + ")");
69:
70:                 // show JSON in textarea
71:                 document.getElementById("code").value = xhr.responseText;
72:
73:                 // insert quote into DOM
74:                 var div = document.createElement("div");
75:                 var text = document.createTextNode(symbol + ": " + quote.price);
76:                 div.appendChild(text);
77:                 document.getElementById("quotes").appendChild(div);
78:             }
79:             else
80:                 alert("Error with Ajax call!");
81:         }
82:     }
83:     xhr.open("GET", url, true);
84:     xhr.send(null);
85: }
86:
87: // ]]>
88: </script>
89: <title></title>
90: </head>
91: <body>
92: <form onsubmit="quote(); return false;">
93:     Symbol: <input id="symbol" type="text" />
94:     <br /><br />
```

```
95:     <input type="submit" value="Get Quote" />
96: </form>
97: <br /><br />
98: <div id="quotes"></div>
99: <br /><br />
100: <textarea cols="80" id="code" rows="16"></textarea>
101: </body>
102: </html>
```

```
1: <!--
2:
3: ajax12.html
4:
5: Gets stock quote from quote7.php via Ajax, using YUI's Connection Manager,
6: embedding result in page itself.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20:   <head>
21:     <script src="http://yui.yahooapis.com/2.5.0/build/yahoo/yahoo.js"></script>
22:     <script src="http://yui.yahooapis.com/2.5.0/build/event/event.js"></script>
23:     <script src="http://yui.yahooapis.com/2.5.0/build/connection/connection.js"></script>
24:     <script type="text/javascript">
25:       // 
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:           // get symbol
36:           var symbol = document.getElementById("symbol").value;
37:
38:           // construct URL
39:           var url = "quote7.php?symbol=" + symbol;
40:
41:           // make call
42:           YAHOO.util.Connect.asyncRequest("GET", url, { success: handler });
43:         }
44:
45:         /*
46:          * void
47:</pre></div><div data-bbox="131 522 217 547" data-label="Page-Header">ajax12.html<br/>lectures/7/src/</div><div data-bbox="835 522 869 535" data-label="Page-Header">2/2</div><div data-bbox="151 552 589 780" data-label="Text"><pre>48:         * handler()
49:         *
50:         * Handles the Ajax response.
51:         */
52:         function handler(o)
53:         {
54:           // evaluate JSON
55:           var quote = eval("(" + o.responseText + ")");
56:
57:           // insert quote into page
58:           document.getElementById("quote").innerHTML = quote.price;
59:         }
60:
61:       // ]]&gt;
62:     &lt;/script&gt;
63:     &lt;title&gt;&lt;/title&gt;
64:   &lt;/head&gt;
65:   &lt;body&gt;
66:     &lt;form onsubmit="quote(); return false;"&gt;
67:       Symbol: &lt;input id="symbol" type="text" /&gt;
68:       &lt;br /&gt;&lt;br /&gt;
69:       &lt;div id="quote"&gt;&lt;b&gt;quote will appear here!&lt;/b&gt;&lt;/div&gt;
70:       &lt;br /&gt;&lt;br /&gt;
71:       &lt;input type="submit" value="Get Quote" /&gt;
72:     &lt;/form&gt;
73:   &lt;/body&gt;
74: &lt;/html&gt;</pre></div>
```

```
1: <!--
2:
3: ajax2.html
4:
5: Gets stock quote from quotel.php via Ajax, embedding result in an input element.
6:
7: David J. Malan
8: Dan Armendariz
9: Computer Science E-75
10: Harvard Extension School
11:
12: -->
13:
14: <!DOCTYPE html PUBLIC
15:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
16:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
17:
18: <html xmlns="http://www.w3.org/1999/xhtml">
19:   <head>
20:     <script type="text/javascript">
21:       // <![CDATA[
22:
23:         // an XMLHttpRequest
24:         var xhr = null;
25:
26:         /*
27:          * void
28:          * quote()
29:          *
30:          * Gets a quote.
31:          */
32:         function quote()
33:         {
34:           // instantiate XMLHttpRequest object
35:           try
36:           {
37:             xhr = new XMLHttpRequest();
38:           }
39:           catch (e)
40:           {
41:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
42:           }
43:
44:           // handle old browsers
45:           if (xhr == null)
46:           {
47:             alert("Ajax not supported by your browser!");
```

```
48:             return;
49:           }
50:
51:           // construct URL
52:           var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
53:
54:           // get quote
55:           xhr.onreadystatechange = handler;
56:           xhr.open("GET", url, true);
57:           xhr.send(null);
58:         }
59:
60:
61:         /*
62:          * void
63:          * handler()
64:          *
65:          * Handles the Ajax response.
66:          */
67:         function handler()
68:         {
69:           // only handle loaded requests
70:           if (xhr.readyState == 4)
71:           {
72:             if (xhr.status == 200)
73:               document.getElementById("price").value = xhr.responseText;
74:             else
75:               alert("Error with Ajax call!");
76:           }
77:         }
78:
79:       // ]]>
80:     </script>
81:     <title></title>
82:   </head>
83:   <body>
84:     <form onsubmit="quote(); return false;">
85:       Symbol: <input id="symbol" type="text" />
86:       <br />
87:       Price: <input id="price" type="text" />
88:       <br /><br />
89:       <input type="submit" value="Get Quote" />
90:     </form>
91:   </body>
92: </html>
```

```
1: <!--
2:
3: ajax3.html
4:
5: Gets stock quote from quotel.php via Ajax, embedding result in page itself.
6:
7: David J. Malan
8: Dan Armendariz
9: Computer Science E-75
10: Harvard Extension School
11:
12: -->
13:
14: <!DOCTYPE html PUBLIC
15:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
16:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
17:
18: <html xmlns="http://www.w3.org/1999/xhtml">
19:   <head>
20:     <script type="text/javascript">
21:       // <![CDATA[
22:
23:         // an XMLHttpRequest
24:         var xhr = null;
25:
26:         /*
27:          * void
28:          * quote()
29:          *
30:          * Gets a quote.
31:          */
32:         function quote()
33:         {
34:           // instantiate XMLHttpRequest object
35:           try
36:           {
37:             xhr = new XMLHttpRequest();
38:           }
39:           catch (e)
40:           {
41:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
42:           }
43:
44:           // handle old browsers
45:           if (xhr == null)
46:           {
47:             alert("Ajax not supported by your browser!");
```

```
48:             return;
49:           }
50:
51:           // construct URL
52:           var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
53:
54:           // get quote
55:           xhr.onreadystatechange = handler;
56:           xhr.open("GET", url, true);
57:           xhr.send(null);
58:         }
59:
60:
61:         /*
62:          * void
63:          * handler()
64:          *
65:          * Handles the Ajax response.
66:          */
67:         function handler()
68:         {
69:           // only handle loaded requests
70:           if (xhr.readyState == 4)
71:           {
72:             if (xhr.status == 200)
73:               document.getElementById("price").innerHTML = xhr.responseText;
74:             else
75:               alert("Error with Ajax call!");
76:           }
77:         }
78:
79:       // ]]>
80:     </script>
81:   <title></title>
82: </head>
83: <body>
84:   <form onsubmit="quote(); return false;">
85:     Symbol: <input id="symbol" type="text" />
86:     <br />
87:     Price: <span id="price"><b>to be determined</b></span>
88:     <br /><br />
89:     <input type="submit" value="Get Quote" />
90:   </form>
91: </body>
92: </html>
```

```
1: <!--
2:
3: ajax4.html
4:
5: Gets stock quote (plus day's low and high) from quote2.php via Ajax,
6: embedding result in a textarea.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20: <head>
21: <script type="text/javascript">
22:     // 
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:             // instantiate XMLHttpRequest object
36:             try
37:             {
38:                 xhr = new XMLHttpRequest();
39:             }
40:             catch (e)
41:             {
42:                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:             }
44:
45:             // handle old browsers
46:             if (xhr == null)
47:             {</pre></div><div data-bbox="131 522 212 547" data-label="Page-Header">ajax4.html<br/>lectures/7/src/</div><div data-bbox="835 522 869 535" data-label="Page-Header">2/2</div><div data-bbox="151 552 683 939" data-label="Text"><pre>48:                 alert("Ajax not supported by your browser!");
49:                 return;
50:             }
51:
52:             // construct URL
53:             var url = "quote2.php?symbol=" + document.getElementById("symbol").value;
54:
55:             // get quote
56:             xhr.onreadystatechange = handler;
57:             xhr.open("GET", url, true);
58:             xhr.send(null);
59:         }
60:
61:
62:         /*
63:          * void
64:          * handler()
65:          *
66:          * Handles the Ajax response.
67:          */
68:         function handler()
69:         {
70:             // only handle loaded requests
71:             if (xhr.readyState == 4)
72:             {
73:                 if (xhr.status == 200)
74:                     document.getElementById("result").value = xhr.responseText;
75:                 else
76:                     alert("Error with Ajax call!");
77:             }
78:         }
79:
80:     // ]]&gt;
81: &lt;/script&gt;
82: &lt;title&gt;&lt;/title&gt;
83: &lt;/head&gt;
84: &lt;body&gt;
85: &lt;form onsubmit="quote(); return false;"&gt;
86:     Symbol: &lt;input id="symbol" type="text" /&gt;
87:     &lt;br /&gt;
88:     &lt;textarea cols="40" id="result" rows="3"&gt;&lt;/textarea&gt;
89:     &lt;br /&gt;&lt;br /&gt;
90:     &lt;input type="submit" value="Get Quote" /&gt;
91: &lt;/form&gt;
92: &lt;/body&gt;
93: &lt;/html&gt;</pre></div>
```



```
1: <!--
2:
3: ajax5.html
4:
5: Gets stock quote (plus day's low and high) from quote3.php via Ajax,
6: embedding result in page itself.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20: <head>
21: <script type="text/javascript">
22:     // 
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:             // instantiate XMLHttpRequest object
36:             try
37:             {
38:                 xhr = new XMLHttpRequest();
39:             }
40:             catch (e)
41:             {
42:                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:             }
44:
45:             // handle old browsers
46:             if (xhr == null)
47:             {</pre></div><div data-bbox="132 522 212 547" data-label="Page-Header">ajax5.html<br/>lectures/7/src/</div><div data-bbox="835 522 869 535" data-label="Page-Header">2/2</div><div data-bbox="151 552 683 939" data-label="Text"><pre>48:                 alert("Ajax not supported by your browser!");
49:                 return;
50:             }
51:
52:             // construct URL
53:             var url = "quote3.php?symbol=" + document.getElementById("symbol").value;
54:
55:             // get quote
56:             xhr.onreadystatechange = handler;
57:             xhr.open("GET", url, true);
58:             xhr.send(null);
59:         }
60:
61:
62:         /*
63:          * void
64:          * handler()
65:          *
66:          * Handles the Ajax response.
67:          */
68:         function handler()
69:         {
70:             // only handle requests in "loaded" state
71:             if (xhr.readyState == 4)
72:             {
73:                 if (xhr.status == 200)
74:                     document.getElementById("quote").innerHTML = xhr.responseText;
75:                 else
76:                     alert("Error with Ajax call!");
77:             }
78:         }
79:
80:     // ]]&gt;
81: &lt;/script&gt;
82: &lt;title&gt;&lt;/title&gt;
83: &lt;/head&gt;
84: &lt;body&gt;
85: &lt;form onsubmit="quote(); return false;"&gt;
86:     Symbol: &lt;input id="symbol" type="text" /&gt;
87:     &lt;br /&gt;&lt;br /&gt;
88:     &lt;div id="quote"&gt;&lt;b&gt;quote will appear here!&lt;/b&gt;&lt;/div&gt;
89:     &lt;br /&gt;&lt;br /&gt;
90:     &lt;input type="submit" value="Get Quote" /&gt;
91: &lt;/form&gt;
92: &lt;/body&gt;
93: &lt;/html&gt;</pre></div>
```

```
1: <!--
2:
3: ajax6.html
4:
5: Gets stock quote (plus day's low and high) from quote4.php via Ajax,
6: embedding result in page itself after indicating progress with text.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20:   <head>
21:     <script type="text/javascript">
22:       // <![CDATA[
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:           // instantiate XMLHttpRequest object
36:           try
37:           {
38:             xhr = new XMLHttpRequest();
39:           }
40:           catch (e)
41:           {
42:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:           }
44:
45:           // handle old browsers
46:           if (xhr == null)
47:           {
```

```
48:             alert("Ajax not supported by your browser!");
49:             return;
50:           }
51:
52:           // construct URL
53:           var url = "quote4.php?symbol=" + document.getElementById("symbol").value;
54:
55:           // inform user
56:           document.getElementById("quote").innerHTML = "Looking up symbol...";
57:
58:           // get quote
59:           xhr.onreadystatechange = handler;
60:           xhr.open("GET", url, true);
61:           xhr.send(null);
62:         }
63:
64:         /*
65:          * void
66:          * handler()
67:          *
68:          * Handles the Ajax response.
69:          */
70:         function handler()
71:         {
72:           // only handle requests in "loaded" state
73:           if (xhr.readyState == 4)
74:           {
75:             if (xhr.status == 200)
76:               document.getElementById("quote").innerHTML = xhr.responseText;
77:             else
78:               alert("Error with Ajax call!");
79:           }
80:         }
81:       }
82:     // ]]>
83:   </script>
84:   <title></title>
85: </head>
86: <body>
87:   <form onsubmit="quote(); return false;">
88:     Symbol: <input id="symbol" type="text" />
89:     <br /><br />
90:     <div id="quote"><b>quote will appear here!</b></div>
91:     <br /><br />
92:     <input type="submit" value="Get Quote" />
93:   </form>
94:
```

```
95: </body>
96: </html>
```

```
1: <!--
2:
3: ajax7.html
4:
5: Gets stock quote (plus day's low and high) from quote4.php via Ajax,
6: embedding result in page itself after indicating progress with an
7: animated GIF.
8:
9: David J. Malan
10: Dan Armendariz
11: Computer Science E-75
12: Harvard Extension School
13:
14: -->
15:
16: <!DOCTYPE html PUBLIC
17:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
18:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
19:
20: <html xmlns="http://www.w3.org/1999/xhtml">
21:   <head>
22:     <script type="text/javascript">
23:       // <![CDATA[
24:
25:         // an XMLHttpRequest
26:         var xhr = null;
27:
28:         /*
29:          * void
30:          * quote()
31:          *
32:          * Gets a quote.
33:          */
34:         function quote()
35:         {
36:           // instantiate XMLHttpRequest object
37:           try
38:           {
39:             xhr = new XMLHttpRequest();
40:           }
41:           catch (e)
42:           {
43:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
44:           }
45:
46:           // handle old browsers
47:           if (xhr == null)
```

```
48:         {
49:             alert("Ajax not supported by your browser!");
50:             return;
51:         }
52:
53:         // construct URL
54:         var url = "quote4.php?symbol=" + document.getElementById("symbol").value;
55:
56:         // show progress
57:         document.getElementById("progress").style.display = "block";
58:
59:         // get quote
60:         xhr.onreadystatechange = handler;
61:         xhr.open("GET", url, true);
62:         xhr.send(null);
63:     }
64:
65:     /*
66:     * void
67:     * handler()
68:     * Handles the Ajax response.
69:     */
70:     function handler()
71:     {
72:         // only handle requests in "loaded" state
73:         if (xhr.readyState == 4)
74:         {
75:             // hide progress
76:             document.getElementById("progress").style.display = "none";
77:
78:             if (xhr.status == 200)
79:                 document.getElementById("quote").innerHTML = xhr.responseText;
80:             else
81:                 alert("Error with Ajax call!");
82:         }
83:     }
84: }
85:
86: // ]]>
87: </script>
88: <title></title>
89: </head>
90: <body>
91: <form onsubmit="quote(); return false;">
92:     Symbol: <input id="symbol" type="text" />
93:     <br /><br />
94: </form>
```

```
95: <div id="progress" style="display: none;">
96:     
97:     <br /><br />
98: </div>
99: <div id="quote"></div>
100: <br /><br />
101: <input type="submit" value="Get Quote" />
102: </form>
103: </body>
104: </html>
```

```
1: <!--
2:
3: ajax8.html
4:
5: Gets stock quote (plus day's low and high) from quote5.php via Ajax,
6: embedding result in page itself.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20:   <head>
21:     <script type="text/javascript">
22:       // 
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:           // instantiate XMLHttpRequest object
36:           try
37:           {
38:             xhr = new XMLHttpRequest();
39:           }
40:           catch (e)
41:           {
42:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:           }
44:
45:           // handle old browsers
46:           if (xhr == null)
47:           {</pre></div><div data-bbox="131 522 213 547" data-label="Page-Header">ajax8.html<br/>lectures/7/src/</div><div data-bbox="835 522 867 535" data-label="Page-Header">2/3</div><div data-bbox="151 552 683 948" data-label="Text"><pre>48:             alert("Ajax not supported by your browser!");
49:             return;
50:           }
51:
52:           // construct URL
53:           var url = "quote5.php?symbol=" + document.getElementById("symbol").value;
54:
55:           // get quote
56:           xhr.onreadystatechange = handler;
57:           xhr.open("GET", url, true);
58:           xhr.send(null);
59:         }
60:
61:
62:         /*
63:          * void
64:          * handler()
65:          *
66:          * Handles the Ajax response.
67:          */
68:         function handler()
69:         {
70:           // only handle requests in "loaded" state
71:           if (xhr.readyState == 4)
72:           {
73:             if (xhr.status == 200)
74:             {
75:               // get XML
76:               var xml = xhr.responseXML;
77:
78:               // update price
79:               var prices = xml.getElementsByTagName("price");
80:               if (prices.length == 1)
81:               {
82:                 var price = prices[0].firstChild.nodeValue;
83:                 document.getElementById("price").innerHTML = price;
84:               }
85:
86:               // update low
87:               var lows = xml.getElementsByTagName("low");
88:               if (lows.length == 1)
89:               {
90:                 var low = lows[0].firstChild.nodeValue;
91:                 document.getElementById("low").innerHTML = low;
92:               }
93:
94:               // update high</pre></div>
```

```
95:             var highs = xml.getElementsByTagName("high");
96:             if (highs.length == 1)
97:             {
98:                 var high = highs[0].firstChild.nodeValue;
99:                 document.getElementById("high").innerHTML = high;
100:             }
101:         }
102:     } else
103:     {
104:         alert("Error with Ajax call!");
105:     }
106: }
107: // ]]>
108: </script>
109: <title></title>
110: </head>
111: <body>
112:     <form onsubmit="quote(); return false;">
113:         Symbol: <input id="symbol" type="text" />
114:         <br /><br />
115:         Price: <span id="price"></span>
116:         <br />
117:         Low: <span id="low"></span>
118:         <br />
119:         High: <span id="high"></span>
120:         <br /><br />
121:         <input type="submit" value="Get Quote" />
122:     </form>
123: </body>
124: </html>
```

```
1: <!--
2:
3: ajax9.html
4:
5: Gets stock quote from quotel.php via Ajax, inserting result into DOM.
6: Also displays new DOM nodes' XHTML in a textarea.
7:
8: David J. Malan
9: Dan Armendariz
10: Computer Science E-75
11: Harvard Extension School
12:
13: -->
14:
15: <!DOCTYPE html PUBLIC
16:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
17:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
18:
19: <html xmlns="http://www.w3.org/1999/xhtml">
20: <head>
21: <script type="text/javascript">
22:     // <![CDATA[
23:
24:         // an XMLHttpRequest
25:         var xhr = null;
26:
27:         /*
28:          * void
29:          * quote()
30:          *
31:          * Gets a quote.
32:          */
33:         function quote()
34:         {
35:             // instantiate XMLHttpRequest object
36:             try
37:             {
38:                 xhr = new XMLHttpRequest();
39:             }
40:             catch (e)
41:             {
42:                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
43:             }
44:
45:             // handle old browsers
46:             if (xhr == null)
47:             {
```

```
48:         alert("Ajax not supported by your browser!");
49:         return;
50:     }
51:
52:     // get symbol
53:     var symbol = document.getElementById("symbol").value;
54:
55:     // construct URL
56:     var url = "quote1.php?symbol=" + symbol;
57:
58:     // get quote
59:     xhr.onreadystatechange =
60:     function()
61:     {
62:         // only handle loaded requests
63:         if (xhr.readyState == 4)
64:         {
65:             if (xhr.status == 200)
66:             {
67:                 // insert quote into DOM
68:                 var div = document.createElement("div");
69:                 var text = document.createTextNode(symbol + ": " + xhr.responseText);
70:                 div.appendChild(text);
71:                 document.getElementById("quotes").appendChild(div);
72:             }
73:             else
74:                 alert("Error with Ajax call!");
75:         }
76:     }
77:     xhr.open("GET", url, true);
78:     xhr.send(null);
79: }
80:
81: // ]]>
82: </script>
83: <title></title>
84: </head>
85: <body>
86: <form onsubmit="quote(); return false;">
87:     Symbol: <input id="symbol" type="text" />
88:     <br /><br />
89:     <input type="submit" value="Get Quote" />
90: </form>
91: <br /><br />
92: <div id="quotes"></div>
93: </body>
94: </html>
```

```
1: <?php
2:
3: /**
4:  * quote1.php
5:  *
6:  * Outputs price of given symbol as text/html.
7:  *
8:  * David J. Malan
9:  * Dan Armendariz
10:  * Computer Science E-75
11:  * Harvard Extension School
12:  */
13:
14: // get quote
15: $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1", "r");
16: if ($handle !== FALSE)
17: {
18:     $data = fgetcsv($handle);
19:     if ($data !== FALSE && $data[0] == "N/A")
20:         print($data[1]);
21:     fclose($handle);
22: }
23: ?>
```

```
1: <?
2:  /**
3:   * quote2.php
4:   *
5:   * Outputs price, low, and high of given symbol as plain/text.
6:   *
7:   * David J. Malan
8:   * Dan Armendariz
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // send MIME type
14:  header("Content-type: text/plain");
15:
16:  // try to get quote
17:  $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
18:  if ($handle !== FALSE)
19:  {
20:      $data = fgetcsv($handle);
21:      if ($data !== FALSE && $data[0] == "N/A")
22:      {
23:          print("Price: {$data[1]}\n");
24:          print("High: {$data[2]}\n");
25:          print("Low: {$data[3]}");
26:      }
27:      fclose($handle);
28:  }
29: ?>
```

```
1: <?
2:  /**
3:   * quote3.php
4:   *
5:   * Outputs price, low, and high of given symbol as text/html.
6:   *
7:   * David J. Malan
8:   * Dan Armendariz
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // try to get quote
14:  $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
15:  if ($handle !== FALSE)
16:  {
17:      $data = fgetcsv($handle);
18:      if ($data !== FALSE && $data[0] == "N/A")
19:      {
20:          print("Price: {$data[1]}");
21:          print("<br />");
22:          print("High: {$data[2]}");
23:          print("<br />");
24:          print("Low: {$data[3]}");
25:      }
26:      fclose($handle);
27:  }
28: ?>
```



```
1: <?
2:  /**
3:   * quote3.php
4:   *
5:   * Outputs price, low, and high of given symbol as text/html, after
6:   * inserting an artificial delay.
7:   *
8:   * David J. Malan
9:   * Dan Armendariz
10:  * Computer Science E-75
11:  * Harvard Extension School
12:  */
13:
14:
15:  // pretend server is slow
16:  sleep(5);
17:
18:  // try to get quote
19:  $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
20:  if ($handle !== FALSE)
21:  {
22:      $data = fgetcsv($handle);
23:      if ($data !== FALSE && $data[0] == "N/A")
24:      {
25:          print("Price: {$data[1]}");
26:          print("<br />");
27:          print("High: {$data[2]}");
28:          print("<br />");
29:          print("Low: {$data[3]}");
30:      }
31:      fclose($handle);
32:  }
33: ?>
```

```
1: <?
2:  /**
3:   * quote5.php
4:   *
5:   * Outputs price, low, and high of given symbol as text/xml.
6:   *
7:   * David J. Malan
8:   * Dan Armendariz
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // set MIME type
14:  header("Content-type: text/xml");
15:
16:  // output root element's start tag
17:  print("<quote symbol='{$_GET['symbol']}'>");
18:
19:  // try to get quote
20:  $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
21:  if ($handle !== FALSE)
22:  {
23:      $data = fgetcsv($handle);
24:      if ($data !== FALSE && $data[0] == "N/A")
25:      {
26:          print("<price>{$data[1]}</price>");
27:          print("<high>{$data[2]}</high>");
28:          print("<low>{$data[3]}</low>");
29:      }
30:      fclose($handle);
31:  }
32:
33:  // output root element's end tag
34:  print("</quote>");
35: ?>
```

```
1: <?
2:  /**
3:   * quote6.php
4:   *
5:   * Outputs price, low, and high of given symbol in JSON format.
6:   *
7:   * David J. Malan
8:   * Dan Armendariz
9:   * Computer Science E-75
10:  * Harvard Extension School
11:  */
12:
13:  // set MIME type
14:  header("Content-type: application/json");
15:
16:  // try to get quote
17:  $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e11hg", "r");
18:  if ($handle !== FALSE)
19:  {
20:      $data = fgetcsv($handle);
21:      if ($data !== FALSE && $data[0] == "N/A")
22:      {
23:          if (is_numeric($data[1]))
24:              $price = $data[1];
25:          if (is_numeric($data[2]))
26:              $high = $data[2];
27:          if (is_numeric($data[3]))
28:              $low = $data[3];
29:      }
30:      fclose($handle);
31:  }
32:
33:  // output JSON
34:  print("{ price: $price, high: $high, low: $low }");
35: ?>
```

```
1: <?
2:  /**
3:   * quote7.php
4:   *
5:   * Outputs price, low, and high of given symbol in JSON format
6:   * using PHP's JSON extension.
7:   *
8:   * David J. Malan
9:   * Dan Armendariz
10:  * Computer Science E-75
11:  * Harvard Extension School
12:  */
13:
14:  // defines a stock
15:  class Stock
16:  {
17:      public $price;
18:      public $high;
19:      public $low;
20:  }
21:
22:  // set MIME type
23:  header("Content-type: application/json");
24:
25:  // try to get quote
26:  $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e11hg", "r");
27:  if ($handle !== FALSE)
28:  {
29:      $data = fgetcsv($handle);
30:      if ($data !== FALSE && $data[0] == "N/A")
31:      {
32:          $stock = new Stock();
33:
34:          if (is_numeric($data[1]))
35:              $stock->price = $data[1];
36:          if (is_numeric($data[2]))
37:              $stock->high = $data[2];
38:          if (is_numeric($data[3]))
39:              $stock->low = $data[3];
40:      }
41:      fclose($handle);
42:  }
43:
44:  // output JSON
45:  print(json_encode($stock));
46: ?>
```