

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

```
48:         }
49:
50:         // construct URL
51:         var url = "quote1.php?symbol=" + document.getElementById("symbol").value;
52:
53:         // get quote
54:         xhr.onreadystatechange = handler;
55:         xhr.open("GET", url, true);
56:         xhr.send(null);
57:     }
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 onsubmit="quote(); return false;">
84:         Symbol: <input id="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: 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:       // 
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!");</pre></div>
```

```
48:         return;
49:     }
50:
51:     // get symbol
52:     var symbol = document.getElementById("symbol").value;
53:
54:     // construct URL
55:     var url = "quote6.php?symbol=" + symbol;
56:
57:     // get quote
58:     xhr.onreadystatechange =
59:     function()
60:     {
61:         // only handle loaded requests
62:         if (xhr.readyState == 4)
63:         {
64:             if (xhr.status == 200)
65:             {
66:                 // evaluate JSON
67:                 var quote = eval("(" + xhr.responseText + ")");
68:
69:                 // show JSON in textarea
70:                 document.getElementById("code").value = xhr.responseText;
71:
72:                 // insert quote into DOM
73:                 var div = document.createElement("div");
74:                 var text = document.createTextNode(symbol + ": " + quote.price);
75:                 div.appendChild(text);
76:                 document.getElementById("quotes").appendChild(div);
77:             }
78:             else
79:                 alert("Error with Ajax call!");
80:         }
81:     }
82:     xhr.open("GET", url, true);
83:     xhr.send(null);
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:         <input type="submit" value="Get Quote" />
```

```
95:      </form>
96:      <br /><br />
97:      <div id="quotes"></div>
98:      <br /><br />
99:      <textarea cols="80" id="code" rows="16"></textarea>
100:    </body>
101: </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: 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:       // 
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!");</pre></div>
```

```
48:         return;
49:     }
50:
51:     // get symbol
52:     var symbol = document.getElementById("symbol").value;
53:
54:     // construct URL
55:     var url = "quote7.php?symbol=" + symbol;
56:
57:     // get quote
58:     xhr.onreadystatechange =
59:     function()
60:     {
61:         // only handle loaded requests
62:         if (xhr.readyState == 4)
63:         {
64:             if (xhr.status == 200)
65:             {
66:                 // evaluate JSON
67:                 var quote = eval("(" + xhr.responseText + ")");
68:
69:                 // show JSON in textarea
70:                 document.getElementById("code").value = xhr.responseText;
71:
72:                 // insert quote into DOM
73:                 var div = document.createElement("div");
74:                 var text = document.createTextNode(symbol + ": " + quote.price);
75:                 div.appendChild(text);
76:                 document.getElementById("quotes").appendChild(div);
77:             }
78:             else
79:                 alert("Error with Ajax call!");
80:         }
81:     }
82:     xhr.open("GET", url, true);
83:     xhr.send(null);
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:         <input type="submit" value="Get Quote" />
```

```
95:      </form>
96:      <br /><br />
97:      <div id="quotes"></div>
98:      <br /><br />
99:      <textarea cols="80" id="code" rows="16"></textarea>
100:    </body>
101: </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: 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 src="http://yui.yahooapis.com/2.5.0/build/yahoo/yahoo.js"></script>
21:     <script src="http://yui.yahooapis.com/2.5.0/build/event/event.js"></script>
22:     <script src="http://yui.yahooapis.com/2.5.0/build/connection/connection.js"></script>
23:     <script type="text/javascript">
24:       // <![CDATA[
25:
26:         /*
27:         * void
28:         * quote()
29:         *
30:         * Gets a quote.
31:         */
32:         function quote()
33:         {
34:           // get symbol
35:           var symbol = document.getElementById("symbol").value;
36:
37:           // construct URL
38:           var url = "quote7.php?symbol=" + symbol;
39:
40:           // make call
41:           YAHOO.util.Connect.asyncRequest("GET", url, { success: handler });
42:         }
43:
44:
45:         /*
46:         * void
47:         * handler()
```

```
48:      *
49:      * Handles the Ajax response.
50:      */
51:      function handler(o)
52:      {
53:          // evaluate JSON
54:          var quote = eval("(" + o.responseText + ")");
55:
56:          // insert quote into page
57:          document.getElementById("quote").innerHTML = quote.price;
58:      }
59:
60:      // ]]>
61:      </script>
62:      <title></title>
63:  </head>
64:  <body>
65:      <form onsubmit="quote(); return false;">
66:          Symbol: <input id="symbol" type="text" />
67:          <br /><br />
68:          <div id="quote"><b>quote will appear here!</b></div>
69:          <br /><br />
70:          <input type="submit" value="Get Quote" />
71:      </form>
72:  </body>
73: </html>
```

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

```
48:         }
49:
50:         // construct URL
51:         var url = "quote1.php?symbol=" + document.getElementById("symbol").value;
52:
53:         // get quote
54:         xhr.onreadystatechange = handler;
55:         xhr.open("GET", url, true);
56:         xhr.send(null);
57:     }
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:                 document.getElementById("price").value = 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 onsubmit="quote(); return false;">
84:         Symbol: <input id="symbol" type="text" />
85:         <br />
86:         Price: <input id="price" type="text" />
87:         <br /><br />
88:         <input type="submit" value="Get Quote" />
89:     </form>
90: </body>
91: </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: Computer Science E-75
9: Harvard Extension School
10:
11: -->
12:
13: <!DOCTYPE html PUBLIC
14:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
15:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
16:
17: <html xmlns="http://www.w3.org/1999/xhtml">
18:   <head>
19:     <script type="text/javascript">
20:       // 
21:
22:         // an XMLHttpRequest
23:         var xhr = null;
24:
25:         /*
26:          * void
27:          * quote()
28:          *
29:          * Gets a quote.
30:          */
31:         function quote()
32:         {
33:           // instantiate XMLHttpRequest object
34:           try
35:           {
36:             xhr = new XMLHttpRequest();
37:           }
38:           catch (e)
39:           {
40:             xhr = new ActiveXObject("Microsoft.XMLHTTP");
41:           }
42:
43:           // handle old browsers
44:           if (xhr == null)
45:           {
46:             alert("Ajax not supported by your browser!");
47:             return;</pre></div>
```

```
48:         }
49:
50:         // construct URL
51:         var url = "quote1.php?symbol=" + document.getElementById("symbol").value;
52:
53:         // get quote
54:         xhr.onreadystatechange = handler;
55:         xhr.open("GET", url, true);
56:         xhr.send(null);
57:     }
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:                 document.getElementById("price").innerHTML = 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 onsubmit="quote(); return false;">
84:         Symbol: <input id="symbol" type="text" />
85:         <br />
86:         Price: <span id="price"><b>to be determined</b></span>
87:         <br /><br />
88:         <input type="submit" value="Get Quote" />
89:     </form>
90: </body>
91: </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: 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:       // 
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!");</pre></div>
```

```
48:         return;
49:     }
50:
51:     // construct URL
52:     var url = "quote2.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("result").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:         <textarea cols="40" id="result" rows="3"></textarea>
88:         <br /><br />
89:         <input type="submit" value="Get Quote" />
90:     </form>
91: </body>
92: </html>
```



```
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: 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:       // 
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!");</pre></div>
```

```
48:         return;
49:     }
50:
51:     // construct URL
52:     var url = "quote3.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 requests in "loaded" state
70:     if (xhr.readyState == 4)
71:     {
72:         if (xhr.status == 200)
73:             document.getElementById("quote").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 /><br />
87:         <div id="quote"><b>quote will appear here!</b></div>
88:         <br /><br />
89:         <input type="submit" value="Get Quote" />
90:     </form>
91: </body>
92: </html>
```

```
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: 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:       // 
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!");</pre></div>
```

```
48:         return;
49:     }
50:
51:     // construct URL
52:     var url = "quote4.php?symbol=" + document.getElementById("symbol").value;
53:
54:     // inform user
55:     document.getElementById("quote").innerHTML = "Looking up symbol...";
56:
57:     // get quote
58:     xhr.onreadystatechange = handler;
59:     xhr.open("GET", url, true);
60:     xhr.send(null);
61: }
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: </body>
```

95: </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: 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>
```

```
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:     // show progress
56:     document.getElementById("progress").style.display = "block";
57:
58:     // get quote
59:     xhr.onreadystatechange = handler;
60:     xhr.open("GET", url, true);
61:     xhr.send(null);
62: }
63:
64:
65: /*
66:  * void
67:  * handler()
68:  *
69:  * Handles the Ajax response.
70:  */
71: function handler()
72: {
73:     // only handle requests in "loaded" state
74:     if (xhr.readyState == 4)
75:     {
76:         // hide progress
77:         document.getElementById("progress").style.display = "none";
78:
79:         if (xhr.status == 200)
80:             document.getElementById("quote").innerHTML = xhr.responseText;
81:         else
82:             alert("Error with Ajax call!");
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:         <div id="progress" style="display: none;">
```

```
95:         
96:         <br /><br />
97:     </div>
98:     <div id="quote"></div>
99:     <br /><br />
100:     <input type="submit" value="Get Quote" />
101: </form>
102: </body>
103: </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: 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:       // 
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!");</pre></div>
```

```
48:         return;
49:     }
50:
51:     // construct URL
52:     var url = "quote5.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 requests in "loaded" state
70:     if (xhr.readyState == 4)
71:     {
72:         if (xhr.status == 200)
73:         {
74:             // get XML
75:             var xml = xhr.responseXML;
76:
77:             // update price
78:             var prices = xml.getElementsByTagName("price");
79:             if (prices.length == 1)
80:             {
81:                 var price = prices[0].firstChild.nodeValue;
82:                 document.getElementById("price").innerHTML = price;
83:             }
84:
85:             // update low
86:             var lows = xml.getElementsByTagName("low");
87:             if (lows.length == 1)
88:             {
89:                 var low = lows[0].firstChild.nodeValue;
90:                 document.getElementById("low").innerHTML = low;
91:             }
92:
93:             // update high
94:             var highs = xml.getElementsByTagName("high");
```

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

```
48:         return;
49:     }
50:
51:     // get symbol
52:     var symbol = document.getElementById("symbol").value;
53:
54:     // construct URL
55:     var url = "quote1.php?symbol=" + symbol;
56:
57:     // get quote
58:     xhr.onreadystatechange =
59:     function()
60:     {
61:         // only handle loaded requests
62:         if (xhr.readyState == 4)
63:         {
64:             if (xhr.status == 200)
65:             {
66:                 // insert quote into DOM
67:                 var div = document.createElement("div");
68:                 var text = document.createTextNode(symbol + ": " + xhr.responseText);
69:                 div.appendChild(text);
70:                 document.getElementById("quotes").appendChild(div);
71:
72:                 // show corresponding XHTML in textarea
73:                 document.getElementById("code").value = document.body.innerHTML;
74:             }
75:             else
76:                 alert("Error with Ajax call!");
77:         }
78:     }
79:     xhr.open("GET", url, true);
80:     xhr.send(null);
81: }
82:
83: // ]]>
84: </script>
85: <title></title>
86: </head>
87: <body>
88:     <form onsubmit="quote(); return false;">
89:         Symbol: <input id="symbol" type="text" />
90:         <br /><br />
91:         <input type="submit" value="Get Quote" />
92:     </form>
93:     <br /><br />
94:     <div id="quotes"></div>
```

```
95:      <br /><br />
96:      <textarea cols="80" id="code" rows="16"></textarea>
97:    </body>
98: </html>
```

```
1: <?
2:     /**
3:      * quote1.php
4:      *
5:      * Outputs price of given symbol as text/html.
6:      *
7:      * David J. Malan
8:      * Computer Science E-75
9:      * Harvard Extension School
10:     */
11:
12:     // get quote
13:     $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e111", "r");
14:     if ($handle !== FALSE)
15:     {
16:         $data = fgetcsv($handle);
17:         if ($data !== FALSE && $data[0] == "N/A")
18:             print($data[1]);
19:         fclose($handle);
20:     }
21: ?>
```

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



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

```
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:      * Computer Science E-75
10:     * Harvard Extension School
11:     */
12:
13:
14:     // pretend server is slow
15:     sleep(5);
16:
17:     // try to get quote
18:     $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
19:     if ($handle !== FALSE)
20:     {
21:         $data = fgetcsv($handle);
22:         if ($data !== FALSE && $data[0] == "N/A")
23:         {
24:             print("Price: {$data[1]}");
25:             print("<br />");
26:             print("High: {$data[2]}");
27:             print("<br />");
28:             print("Low: {$data[3]}");
29:         }
30:         fclose($handle);
31:     }
32: ?>
```

```
1: <?
2:     /**
3:      * quote5.php
4:      *
5:      * Outputs price, low, and high of given symbol as text/xml.
6:      *
7:      * David J. Malan
8:      * Computer Science E-75
9:      * Harvard Extension School
10:     */
11:
12:     // set MIME type
13:     header("Content-type: text/xml");
14:
15:     // output root element's start tag
16:     print("<quote symbol='{$_GET['symbol']}'>");
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]}</price>");
26:             print("<high>{$data[2]}</high>");
27:             print("<low>{$data[3]}</low>");
28:         }
29:         fclose($handle);
30:     }
31:
32:     // output root element's end tag
33:     print("</quote>");
34: ?>
```

```
1: <?
2:     /**
3:      * quote6.php
4:      *
5:      * Outputs price, low, and high of given symbol in JSON format.
6:      *
7:      * David J. Malan
8:      * Computer Science E-75
9:      * Harvard Extension School
10:     */
11:
12:     // set MIME type
13:     header("Content-type: application/json");
14:
15:     // try to get quote
16:     $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
17:     if ($handle !== FALSE)
18:     {
19:         $data = fgetcsv($handle);
20:         if ($data !== FALSE && $data[0] == "N/A")
21:         {
22:             if (is_numeric($data[1]))
23:                 $price = $data[1];
24:             if (is_numeric($data[2]))
25:                 $high = $data[2];
26:             if (is_numeric($data[3]))
27:                 $low = $data[3];
28:         }
29:         fclose($handle);
30:     }
31:
32:     // output JSON
33:     print("{ price: $price, high: $high, low: $low }");
34: ?>
```

```
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:      * Computer Science E-75
10:     * Harvard Extension School
11:     */
12:
13:     // defines a stock
14:     class Stock
15:     {
16:         public $price;
17:         public $high;
18:         public $low;
19:     }
20:
21:     // set MIME type
22:     header("Content-type: application/json");
23:
24:     // try to get quote
25:     $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e1l1hg", "r");
26:     if ($handle !== FALSE)
27:     {
28:         $data = fgetcsv($handle);
29:         if ($data !== FALSE && $data[0] == "N/A")
30:         {
31:             $stock = new Stock();
32:
33:             if (is_numeric($data[1]))
34:                 $stock->price = $data[1];
35:             if (is_numeric($data[2]))
36:                 $stock->high = $data[2];
37:             if (is_numeric($data[3]))
38:                 $stock->low = $data[3];
39:         }
40:         fclose($handle);
41:     }
42:
43:     // output JSON
44:     print(json_encode($stock));
45: ?>
```