

```
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 S-75
9. Harvard Summer School
10.
11. -->
12.
13. <!DOCTYPE html>
14.
15. <html>
16.   <head>
17.     <script>
18.
19.         // an XMLHttpRequest
20.         var xhr = null;
21.
22.         /*
23.          * void
24.          * quote()
25.          *
26.          * Gets a quote.
27.          */
28.         function quote()
29.         {
30.             // instantiate XMLHttpRequest object
31.             try
32.             {
33.                 xhr = new XMLHttpRequest();
34.             }
35.             catch (e)
36.             {
37.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
38.             }
39.
40.             // handle old browsers
41.             if (xhr == null)
42.             {
43.                 alert("Ajax not supported by your browser!");
44.                 return;
45.             }
46.
47.             // construct URL
48.             var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
```

```
49.
50.     // get quote
51.     xhr.onreadystatechange = handler;
52.     xhr.open("GET", url, true);
53.     xhr.send(null);
54. }
55.
56.
57. /*
58.  * void
59.  * handler()
60.  *
61.  * Handles the Ajax response.
62.  */
63. function handler()
64. {
65.     // only handle loaded requests
66.     if (xhr.readyState == 4)
67.     {
68.         if (xhr.status == 200)
69.             alert(xhr.responseText);
70.         else
71.             alert("Error with Ajax call!");
72.     }
73. }
74.
75. </script>
76. <title></title>
77. </head>
78. <body>
79.     <form action="quotel.php" method="get" onsubmit="quote(); return false;">
80.         Symbol: <input id="symbol" name="symbol" type="text">
81.         <br><br>
82.         <input type="submit" value="Get Quote">
83.     </form>
84. </body>
85. </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 S-75
9. Harvard Summer School
10.
11. -->
12.
13. <!DOCTYPE html>
14.
15. <html>
16.   <head>
17.     <script>
18.
19.         // an XMLHttpRequest
20.         var xhr = null;
21.
22.         /*
23.          * void
24.          * quote()
25.          *
26.          * Gets a quote.
27.          */
28.         function quote()
29.         {
30.             // instantiate XMLHttpRequest object
31.             try
32.             {
33.                 xhr = new XMLHttpRequest();
34.             }
35.             catch (e)
36.             {
37.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
38.             }
39.
40.             // handle old browsers
41.             if (xhr == null)
42.             {
43.                 alert("Ajax not supported by your browser!");
44.                 return;
45.             }
46.
47.             // construct URL
48.             var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
```

```
49.
50.     // get quote
51.     xhr.onreadystatechange = handler;
52.     xhr.open("GET", url, true);
53.     xhr.send(null);
54. }
55.
56.
57. /*
58.  * void
59.  * handler()
60.  *
61.  * Handles the Ajax response.
62.  */
63. function handler()
64. {
65.     // only handle loaded requests
66.     if (xhr.readyState == 4)
67.     {
68.         if (xhr.status == 200)
69.             document.getElementById("price").value = xhr.responseText;
70.         else
71.             alert("Error with Ajax call!");
72.     }
73. }
74.
75. </script>
76. <title></title>
77. </head>
78. <body>
79.     <form onsubmit="quote(); return false;">
80.         Symbol: <input id="symbol" type="text">
81.         <br>
82.         Price: <input id="price" type="text">
83.         <br><br>
84.         <input type="submit" value="Get Quote">
85.     </form>
86. </body>
87. </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 S-75
9. Harvard Summer School
10.
11. -->
12.
13. <!DOCTYPE html>
14.
15. <html>
16.   <head>
17.     <script>
18.
19.         // an XMLHttpRequest
20.         var xhr = null;
21.
22.         /*
23.          * void
24.          * quote()
25.          *
26.          * Gets a quote.
27.          */
28.         function quote()
29.         {
30.             // instantiate XMLHttpRequest object
31.             try
32.             {
33.                 xhr = new XMLHttpRequest();
34.             }
35.             catch (e)
36.             {
37.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
38.             }
39.
40.             // handle old browsers
41.             if (xhr == null)
42.             {
43.                 alert("Ajax not supported by your browser!");
44.                 return;
45.             }
46.
47.             // construct URL
48.             var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
```

```
49.
50.     // get quote
51.     xhr.onreadystatechange = handler;
52.     xhr.open("GET", url, true);
53.     xhr.send(null);
54. }
55.
56.
57. /*
58.  * void
59.  * handler()
60.  *
61.  * Handles the Ajax response.
62.  */
63. function handler()
64. {
65.     // only handle loaded requests
66.     if (xhr.readyState == 4)
67.     {
68.         if (xhr.status == 200)
69.             document.getElementById("price").innerHTML = xhr.responseText;
70.         else
71.             alert("Error with Ajax call!");
72.     }
73. }
74.
75. </script>
76. <title></title>
77. </head>
78. <body>
79.     <form onsubmit="quote(); return false;">
80.         Symbol: <input id="symbol" type="text">
81.         <br>
82.         Price: <span id="price"><b>to be determined</b></span>
83.         <br><br>
84.         <input type="submit" value="Get Quote">
85.     </form>
86. </body>
87. </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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // construct URL
```

```
49.         var url = "quote2.php?symbol=" + document.getElementById("symbol").value;
50.
51.         // get quote
52.         xhr.onreadystatechange = handler;
53.         xhr.open("GET", url, true);
54.         xhr.send(null);
55.     }
56.
57.
58.     /*
59.     * void
60.     * handler()
61.     *
62.     * Handles the Ajax response.
63.     */
64.     function handler()
65.     {
66.         // only handle loaded requests
67.         if (xhr.readyState == 4)
68.         {
69.             if (xhr.status == 200)
70.                 document.getElementById("result").value = xhr.responseText;
71.             else
72.                 alert("Error with Ajax call!");
73.         }
74.     }
75.
76.     </script>
77.     <title></title>
78. </head>
79. <body>
80.     <form onsubmit="quote(); return false;">
81.         Symbol: <input id="symbol" type="text">
82.         <br>
83.         <textarea cols="40" id="result" rows="3"></textarea>
84.         <br><br>
85.         <input type="submit" value="Get Quote">
86.     </form>
87. </body>
88. </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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // construct URL
```

```
49.         var url = "quote3.php?symbol=" + document.getElementById("symbol").value;
50.
51.         // get quote
52.         xhr.onreadystatechange = handler;
53.         xhr.open("GET", url, true);
54.         xhr.send(null);
55.     }
56.
57.
58.     /*
59.     * void
60.     * handler()
61.     *
62.     * Handles the Ajax response.
63.     */
64.     function handler()
65.     {
66.         // only handle requests in "loaded" state
67.         if (xhr.readyState == 4)
68.         {
69.             if (xhr.status == 200)
70.                 document.getElementById("quote").innerHTML = xhr.responseText;
71.             else
72.                 alert("Error with Ajax call!");
73.         }
74.     }
75.
76.     </script>
77.     <title></title>
78. </head>
79. <body>
80.     <form onsubmit="quote(); return false;">
81.         Symbol: <input id="symbol" type="text">
82.         <br><br>
83.         <div id="quote"><b>quote will appear here!</b></div>
84.         <br><br>
85.         <input type="submit" value="Get Quote">
86.     </form>
87. </body>
88. </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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // construct URL
```

```
49.         var url = "quote4.php?symbol=" + document.getElementById("symbol").value;
50.
51.         // inform user
52.         document.getElementById("quote").innerHTML = "Looking up symbol...";
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.     </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.         <div id="quote"><b>quote will appear here!</b></div>
87.         <br><br>
88.         <input type="submit" value="Get Quote">
89.     </form>
90. </body>
91. </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 S-75
11. Harvard Summer School
12.
13. -->
14.
15. <!DOCTYPE html>
16.
17. <html>
18.   <head>
19.     <script>
20.
21.         // an XMLHttpRequest
22.         var xhr = null;
23.
24.         /*
25.          * void
26.          * quote()
27.          *
28.          * Gets a quote.
29.          */
30.         function quote()
31.         {
32.             // instantiate XMLHttpRequest object
33.             try
34.             {
35.                 xhr = new XMLHttpRequest();
36.             }
37.             catch (e)
38.             {
39.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
40.             }
41.
42.             // handle old browsers
43.             if (xhr == null)
44.             {
45.                 alert("Ajax not supported by your browser!");
46.                 return;
47.             }
48.
```

```
49.         // construct URL
50.         var url = "quote4.php?symbol=" + document.getElementById("symbol").value;
51.
52.         // show progress
53.         document.getElementById("progress").style.display = "block";
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.             // hide progress
74.             document.getElementById("progress").style.display = "none";
75.
76.             if (xhr.status == 200)
77.                 document.getElementById("quote").innerHTML = xhr.responseText;
78.             else
79.                 alert("Error with Ajax call!");
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="progress" style="display: none;">
91.             
92.             <br><br>
93.         </div>
94.         <div id="quote"></div>
95.         <br><br>
96.         <input type="submit" value="Get Quote">
```

---

```
97.     </form>
98. </body>
99. </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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // construct URL
```



```
49.         var url = "quote5.php?symbol=" + document.getElementById("symbol").value;
50.
51.         // get quote
52.         xhr.onreadystatechange = handler;
53.         xhr.open("GET", url, true);
54.         xhr.send(null);
55.     }
56.
57.
58.     /*
59.     * void
60.     * handler()
61.     *
62.     * Handles the Ajax response.
63.     */
64.     function handler()
65.     {
66.         // only handle requests in "loaded" state
67.         if (xhr.readyState == 4)
68.         {
69.             if (xhr.status == 200)
70.             {
71.                 // get XML
72.                 var xml = xhr.responseXML;
73.
74.                 // update price
75.                 var prices = xml.getElementsByTagName("price");
76.                 if (prices.length == 1)
77.                 {
78.                     var price = prices[0].firstChild.nodeValue;
79.                     document.getElementById("price").innerHTML = price;
80.                 }
81.
82.                 // update low
83.                 var lows = xml.getElementsByTagName("low");
84.                 if (lows.length == 1)
85.                 {
86.                     var low = lows[0].firstChild.nodeValue;
87.                     document.getElementById("low").innerHTML = low;
88.                 }
89.
90.                 // update high
91.                 var highs = xml.getElementsByTagName("high");
92.                 if (highs.length == 1)
93.                 {
94.                     var high = highs[0].firstChild.nodeValue;
95.                     document.getElementById("high").innerHTML = high;
96.                 }
97.             }
98.         }
99.     }
100.
```

```
97.         }
98.         else
99.             alert("Error with Ajax call!");
100.     }
101. }
102.
103. </script>
104. <title></title>
105. </head>
106. <body>
107.     <form onsubmit="quote(); return false;">
108.         Symbol: <input id="symbol" type="text">
109.         <br><br>
110.         Price: <span id="price"></span>
111.         <br>
112.         Low: <span id="low"></span>
113.         <br>
114.         High: <span id="high"></span>
115.         <br><br>
116.         <input type="submit" value="Get Quote">
117.     </form>
118. </body>
119. </html>
```

```
1. <!--
2.
3. ajax9.html
4.
5. Gets stock quote from quote1.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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // get symbol
```

```
49.         var symbol = document.getElementById("symbol").value;
50.
51.         // construct URL
52.         var url = "quotel.php?symbol=" + symbol;
53.
54.         // get quote
55.         xhr.onreadystatechange =
56.         function()
57.         {
58.             // only handle loaded requests
59.             if (xhr.readyState == 4)
60.             {
61.                 if (xhr.status == 200)
62.                 {
63.                     // insert quote into DOM
64.                     var div = document.createElement("div");
65.                     var text = document.createTextNode(symbol + ": " + xhr.responseText);
66.                     div.appendChild(text);
67.                     document.getElementById("quotes").appendChild(div);
68.                 }
69.                 else
70.                     alert("Error with Ajax call!");
71.             }
72.         }
73.         xhr.open("GET", url, true);
74.         xhr.send(null);
75.     }
76.
77.     </script>
78.     <title></title>
79. </head>
80. <body>
81.     <form onsubmit="quote(); return false;">
82.         Symbol: <input id="symbol" type="text">
83.         <br><br>
84.         <input type="submit" value="Get Quote">
85.     </form>
86.     <br><br>
87.     <div id="quotes"></div>
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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // get symbol
```

```
49.         var symbol = document.getElementById("symbol").value;
50.
51.         // construct URL
52.         var url = "quote6.php?symbol=" + symbol;
53.
54.         // get quote
55.         xhr.onreadystatechange =
56.         function()
57.         {
58.             // only handle loaded requests
59.             if (xhr.readyState == 4)
60.             {
61.                 if (xhr.status == 200)
62.                 {
63.                     // evaluate JSON
64.                     var quote = eval("(" + xhr.responseText + ")");
65.
66.                     // show JSON in textarea
67.                     document.getElementById("code").value = xhr.responseText;
68.
69.                     // insert quote into DOM
70.                     var div = document.createElement("div");
71.                     var text = document.createTextNode(symbol + ": " + quote.price);
72.                     div.appendChild(text);
73.                     document.getElementById("quotes").appendChild(div);
74.                 }
75.                 else
76.                     alert("Error with Ajax call!");
77.             }
78.         }
79.         xhr.open("GET", url, true);
80.         xhr.send(null);
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.         <input type="submit" value="Get Quote">
91.     </form>
92.     <br><br>
93.     <div id="quotes"></div>
94.     <br><br>
95.     <textarea cols="80" id="code" rows="16"></textarea>
96. </body>
```

97. `</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 S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script>
19.
20.         // an XMLHttpRequest
21.         var xhr = null;
22.
23.         /*
24.          * void
25.          * quote()
26.          *
27.          * Gets a quote.
28.          */
29.         function quote()
30.         {
31.             // instantiate XMLHttpRequest object
32.             try
33.             {
34.                 xhr = new XMLHttpRequest();
35.             }
36.             catch (e)
37.             {
38.                 xhr = new ActiveXObject("Microsoft.XMLHTTP");
39.             }
40.
41.             // handle old browsers
42.             if (xhr == null)
43.             {
44.                 alert("Ajax not supported by your browser!");
45.                 return;
46.             }
47.
48.             // get symbol
```



```
49.         var symbol = document.getElementById("symbol").value;
50.
51.         // construct URL
52.         var url = "quote7.php?symbol=" + symbol;
53.
54.         // get quote
55.         xhr.onreadystatechange =
56.         function()
57.         {
58.             // only handle loaded requests
59.             if (xhr.readyState == 4)
60.             {
61.                 if (xhr.status == 200)
62.                 {
63.                     // evaluate JSON
64.                     var quote = eval("(" + xhr.responseText + ")");
65.
66.                     // show JSON in textarea
67.                     document.getElementById("code").value = xhr.responseText;
68.
69.                     // insert quote into DOM
70.                     var div = document.createElement("div");
71.                     var text = document.createTextNode(symbol + ": " + quote.price);
72.                     div.appendChild(text);
73.                     document.getElementById("quotes").appendChild(div);
74.                 }
75.                 else
76.                     alert("Error with Ajax call!");
77.             }
78.         }
79.         xhr.open("GET", url, true);
80.         xhr.send(null);
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.         <input type="submit" value="Get Quote">
91.     </form>
92.     <br><br>
93.     <div id="quotes"></div>
94.     <br><br>
95.     <textarea cols="80" id="code" rows="16"></textarea>
96. </body>
```

97. `</html>`

```
1. <!--
2.
3. ajax12.html
4.
5. Gets stock quote (plus day's low and high) from quote7.php via Ajax
6. using jQuery, embedding (JSON) result in page itself.
7.
8. David J. Malan
9. Computer Science S-75
10. Harvard Summer School
11.
12. -->
13.
14. <!DOCTYPE html>
15.
16. <html>
17.   <head>
18.     <script src="http://code.jquery.com/jquery-latest.js"></script>
19.     <script>
20.
21.       $(document).ready(function() {
22.         $("#form").submit(function() {
23.           $.ajax({
24.             url: "quote7.php",
25.             data: {
26.               symbol: $("#symbol").val()
27.             },
28.             success: function(data) {
29.               $("#price").html(data.price);
30.               $("#high").html(data.high);
31.               $("#low").html(data.low);
32.             }
33.           });
34.           return false;
35.         });
36.       });
37.
38.     </script>
39.     <title></title>
40.   </head>
41.   <body>
42.     <form id="form">
43.       Symbol: <input id="symbol" type="text">
44.       <br><br>
45.       Price: <span id="price"></span>
46.       <br>
47.       High: <span id="high"></span>
48.       <br>
```

```
49.     Low: <span id="low"></span>
50.     <br>
51.     <br><br>
52.     <input type="submit" value="Get Quote">
53. </form>
54. </body>
55. </html>
```

```
1. <?php
2.
3.     /**
4.      * quote1.php
5.      *
6.      * Outputs price of given symbol as text/html.
7.      *
8.      * David J. Malan
9.      * Computer Science S-75
10.     * Harvard Summer School
11.     */
12.
13.     // get quote
14.     $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET['symbol']}&f=e111", "r");
15.     if ($handle !== FALSE)
16.     {
17.         $data = fgetcsv($handle);
18.         if ($data !== FALSE && $data[0] == "N/A")
19.             print($data[1]);
20.         fclose($handle);
21.     }
22. ?>
```

```
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 S-75
9.      * Harvard Summer 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]}");
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 S-75
9.      * Harvard Summer 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 S-75
10.     * Harvard Summer 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=e1llhg", "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 S-75
9.      * Harvard Summer 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 S-75
9.      * Harvard Summer 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 S-75
10.     * Harvard Summer 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. ?>
```