

```
1: <!--
2:
3: ajax4.html
4:
5: Gets stock quotes for multiple companies from quote3.php via Ajax,
6: embedding result in page itself using DOM methods.
7:
8: Computer Science 50
9: David J. Malan
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 request = null;
24:
25:         /*
26:          * void
27:          * quote()
28:          *
29:          * Gets a quote.
30:          */
31:         function quote(symbols)
32:         {
33:           // instantiate XMLHttpRequest object
34:           try
35:           {
36:             request = new XMLHttpRequest();
37:           }
38:           catch (e)
39:           {
40:             request = new ActiveXObject("Microsoft.XMLHTTP");
41:           }
42:
43:           // handle old browsers
44:           if (request == null)
45:           {
46:             alert("Ajax not supported by your browser!");
47:             return;</pre></div><div data-bbox="127 527 179 550" data-label="Page-Header">11/29/08<br/>20:36:02</div><div data-bbox="398 539 596 553" data-label="Page-Header">lectures/weeks/10/src/ajax4.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 820 967" data-label="Text"><pre>48:     }
49:
50:     // construct URL
51:     var url = "quote3.php?symbols=" + symbols;
52:
53:     // get quotes
54:     request.onreadystatechange = handler;
55:     request.open("GET", url, true);
56:     request.send(null);
57:   }
58:
59:
60:   /*
61:   * void
62:   * handler()
63:   *
64:   * Handles the Ajax response.
65:   */
66:   function handler()
67:   {
68:     // prepare to generate XHTML
69:     var xhtml = "";
70:
71:     // only handle requests in "loaded" state
72:     if (request.readyState == 4)
73:     {
74:       // ensure request succeeded
75:       if (request.status == 200)
76:       {
77:         // get array of &lt;quote&gt;'s
78:         var quotes = request.responseXML.getElementsByTagName("quote");
79:
80:         // iterate over quotes
81:         for (var i = 0; i &lt; quotes.length; i++)
82:         {
83:           // grab current quote
84:           var quote = quotes[i];
85:
86:           xhtml += "&lt;b&gt;";
87:           xhtml += quote.getAttribute("name");
88:           xhtml += " (" + quote.getAttribute("symbol") + ")";
89:           xhtml += "&lt;/b&gt;";
90:           xhtml += "&lt;br /&gt;&lt;br /&gt;";
91:           xhtml += "Price: $" + quote.getElementsByTagName("price")[0].firstChild.nodeValue;
92:           xhtml += "&lt;br /&gt;";
93:           xhtml += "High: $" + quote.getElementsByTagName("high")[0].firstChild.nodeValue;
94:           xhtml += "&lt;br /&gt;";</pre></div>
```

```
95:             xmlhttp += "Low: $" + quote.getElementsByTagName("low")[0].firstChild.nodeValue;
96:             xmlhttp += "<br /><br />";
97:         }
98:     }
99:
100:    // else report error
101:    else
102:        alert("HTTP error: " + request.status);
103:    }
104:
105:    // insert XHTML into DOM
106:    document.getElementById("quotes").innerHTML = xmlhttp;
107:    }
108:
109:    // ]]>
110: </script>
111: <title></title>
112: </head>
113: <body>
114:     <form action="" onsubmit="quote(this.symbols.value); return false;">
115:         Symbols: <input name="symbols" type="text" />
116:         <br /><br />
117:         <input type="submit" value="Get Quotes" />
118:     </form>
119:     <br /><br />
120: <div id="quotes"></div>
121: </body>
122: </html>
```

```
1: <!--
2:
3: ajax4.html
4:
5: Gets stock quotes for multiple companies from quote3.php via Ajax,
6: embedding result in page itself using DOM methods.
7:
8: Also displays responseText in a textarea for debugging.
9:
10: Computer Science 50
11: David J. Malan
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 request = null;
26:
27:                 /*
28:                  * void
29:                  * quote()
30:                  *
31:                  * Gets a quote.
32:                  */
33:                 function quote(symbols)
34:                 {
35:                     // instantiate XMLHttpRequest object
36:                     try
37:                     {
38:                         request = new XMLHttpRequest();
39:                     }
40:                     catch (e)
41:                     {
42:                         request = new ActiveXObject("Microsoft.XMLHTTP");
43:                     }
44:
45:                     // handle old browsers
46:                     if (request == null)
47:                     {</pre></div>
```

```
48:         alert("Ajax not supported by your browser!");
49:         return;
50:     }
51:
52:     // construct URL
53:     var url = "quote3.php?symbols=" + symbols;
54:
55:     // get quotes
56:     request.onreadystatechange = handler;
57:     request.open("GET", url, true);
58:     request.send(null);
59: }
60:
61:
62: /*
63:  * void
64:  * handler()
65:  *
66:  * Handles the Ajax response.
67:  */
68: function handler()
69: {
70:     // prepare to generate XHTML
71:     var xhtml = "";
72:
73:     // only handle requests in "loaded" state
74:     if (request.readyState == 4)
75:     {
76:         // ensure request succeeded
77:         if (request.status == 200)
78:         {
79:             // output response to debugging window
80:             document.getElementById("debugging").value = request.responseText;
81:
82:             // get array of <quote>'s
83:             var quotes = request.responseXML.getElementsByTagName("quote");
84:
85:             // iterate over quotes
86:             for (var i = 0; i < quotes.length; i++)
87:             {
88:                 // grab current quote
89:                 var quote = quotes[i];
90:
91:                 xhtml += "<b>";
92:                 xhtml += quote.getAttribute("name");
93:                 xhtml += " (" + quote.getAttribute("symbol") + ")";
94:                 xhtml += "</b>";
```

```
95:         xhtml += "<br /><br />";
96:         xhtml += "Price: $" + quote.getElementsByTagName("price")[0].firstChild.nodeValue;
97:         xhtml += "<br />";
98:         xhtml += "High: $" + quote.getElementsByTagName("high")[0].firstChild.nodeValue;
99:         xhtml += "<br />";
100:        xhtml += "Low: $" + quote.getElementsByTagName("low")[0].firstChild.nodeValue;
101:        xhtml += "<br /><br />";
102:    }
103: }
104:
105: // else report error
106: else
107:     alert("HTTP error: " + request.status);
108: }
109:
110: // insert XHTML into DOM
111: document.getElementById("quotes").innerHTML = xhtml;
112: }
113:
114: // ]]>
115: </script>
116: <title></title>
117: </head>
118: <body>
119:     <form action="" onsubmit="quote(this.symbols.value); return false;">
120:         Symbols: <input name="symbols" type="text" />
121:         <br /><br />
122:         <input type="submit" value="Get Quotes" />
123:         <br /><br />
124:         Debugging Window:
125:         <br />
126:         <textarea id="debugging" cols="80" rows="11"></textarea>
127:     </form>
128:     <br /><br />
129:     <div id="quotes"></div>
130: </body>
131: </html>
```

```
1: <!--
2:
3: ajax6.html
4:
5: Gets stock quotes for multiple companies from quote3.php via Ajax,
6: embedding result in page itself using DOM methods.
7:
8: Demonstrates closures.
9:
10: Computer Science 50
11: David J. Malan
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:         /*
25:          * void
26:          * quote()
27:          *
28:          * Gets a quote.
29:          */
30:         function quote(symbols)
31:         {
32:           // an XMLHttpRequest
33:           var request = null;
34:
35:           // instantiate XMLHttpRequest object
36:           try
37:           {
38:             request = new XMLHttpRequest();
39:           }
40:           catch (e)
41:           {
42:             request = new ActiveXObject("Microsoft.XMLHTTP");
43:           }
44:
45:           // handle old browsers
46:           if (request == null)
47:           {</pre></div><div data-bbox="127 527 179 550" data-label="Page-Header">11/24/08<br/>09:26:52</div><div data-bbox="398 539 596 552" data-label="Page-Header">lectures/weeks/10/src/ajax6.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 846 967" data-label="Text"><pre>48:           alert("Ajax not supported by your browser!");
49:           return;
50:         }
51:
52:         // construct URL
53:         var url = "quote3.php?symbols=" + symbols;
54:
55:         // get quotes
56:         request.onreadystatechange = function() {
57:
58:           // prepare to generate XHTML
59:           var xhttp = "";
60:
61:           // only handle requests in "loaded" state
62:           if (request.readyState == 4)
63:           {
64:             // ensure request succeeded
65:             if (request.status == 200)
66:             {
67:               // get array of &lt;quote&gt;'s
68:               var quotes = request.responseXML.getElementsByTagName("quote");
69:
70:               // iterate over quotes
71:               for (var i = 0; i &lt; quotes.length; i++)
72:               {
73:                 // grab current quote
74:                 var quote = quotes[i];
75:
76:                 // generate XHTML
77:                 xhttp += "&lt;b&gt;";
78:                 xhttp += quote.getAttribute("name");
79:                 xhttp += " (" + quote.getAttribute("symbol") + ")";
80:                 xhttp += "&lt;/b&gt;";
81:                 xhttp += "&lt;br /&gt;&lt;br /&gt;";
82:                 xhttp += "Price: $" + quote.getElementsByTagName("price")[0].firstChild.nodeValue;
83:                 xhttp += "&lt;br /&gt;";
84:                 xhttp += "High: $" + quote.getElementsByTagName("high")[0].firstChild.nodeValue;
85:                 xhttp += "&lt;br /&gt;";
86:                 xhttp += "Low: $" + quote.getElementsByTagName("low")[0].firstChild.nodeValue;
87:                 xhttp += "&lt;br /&gt;&lt;br /&gt;";
88:               }
89:             }
90:
91:           // else report error
92:           else
93:             alert("HTTP error: " + request.status);
94:         }
}</pre></div>
```

```
95:
96:         // insert XHTML into DOM
97:         document.getElementById("quotes").innerHTML = xhttp;
98:
99:         // highlight text field
100:        document.getElementById("symbols").focus();
101:        document.getElementById("symbols").select();
102:    }
103:
104:        // send request
105:        request.open("GET", url, true);
106:        request.send(null);
107:    }
108:
109:    // ]]>
110: </script>
111: </title></title>
112: </head>
113: <body>
114:     <form action="" onsubmit="quote(this.symbols.value); return false;">
115:         Symbols: <input id="symbols" name="symbols" type="text" />
116:         <br /><br />
117:         <input type="submit" value="Get Quotes" />
118:     </form>
119:     <br /><br />
120:     <div id="quotes"></div>
121: </body>
122: </html>
```

```
1: /*****
2:  * hello.c
3:  *
4:  * Computer Science 50
5:  * David J. Malan
6:  *
7:  * Says hello to the world.
8:  *****/
9:
10: #include <stdio.h>
11:
12: int
13: main(int argc, char *argv[])
14: {
15:     printf("hello, world\n");
16: }
```

```
1:      .file "hello.c"
2:      .section .rodata
3: .LC0:
4:      .string "hello, world\n"
5:      .text
6: .globl main
7:      .type main, @function
8: main:
9:      pushl %ebp
10:     movl %esp, %ebp
11:     subl $8, %esp
12:     andl $-16, %esp
13:     movl $0, %eax
14:     addl $15, %eax
15:     addl $15, %eax
16:     shrl $4, %eax
17:     sall $4, %eax
18:     subl %eax, %esp
19:     subl $12, %esp
20:     pushl $.LC0
21:     call printf
22:     addl $16, %esp
23:     leave
24:     ret
25:     .size main, .-main
26:     .section .note.GNU-stack,"",@progbits
27:     .ident "GCC: (GNU) 3.4.6 20060404 (Red Hat 3.4.6-10)"
```

```
1: <!--
2:
3: map1.html
4:
5: Demonstrates a "hello, world" of maps.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12:
13: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
14: "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA8igYd929VTmOEMLNjNyP1xQBpXogqfcSB7goB
19: POHTPn2IeSNTRQdD9mEVL94XCoVb0q6KdbS4ouIDA" type="text/javascript"></script>
20:     <script type="text/javascript">
21:       //
22:         function load()
23:         {
24:           if (GBrowserIsCompatible())
25:           {
26:             var map = new GMap2(document.getElementById("map"));
27:             map.setCenter(new GLatLng(37.4419, -122.1419), 13);
28:           }
29:         }
30:
31:       //]]&gt;
32:     &lt;/script&gt;
33:     &lt;title&gt;&lt;/title&gt;
34:   &lt;/head&gt;
35:   &lt;body onload="load()" onunload="GUnload()"&gt;
36:     &lt;div id="map" style="width: 800px; height: 500px"&gt;&lt;/div&gt;
37:   &lt;/body&gt;
38: &lt;/html&gt;
39:</pre></div>
```

```
1: <!--
2:
3: map2.html
4:
5: Demonstrates map that fills the viewport using CSS.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12:
13: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml" style="height: 100%;">
17:   <head>
18:     <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA8igYd929VTmOEMLNjNyPlxQBpXogqfcSB7goB
POHTPn2IeSNTRQdD9mEVL94XCoVb0q6KdbS4ouIDA" type="text/javascript"></script>
19:     <script type="text/javascript">
20:       <![CDATA[
21:
22:         function load()
23:         {
24:             if (GBrowserIsCompatible())
25:             {
26:                 var map = new GMap2(document.getElementById("map"));
27:                 map.setCenter(new GLatLng(37.4419, -122.1419), 13);
28:             }
29:         }
30:
31:       </script>
32:     </script>
33:     <title></title>
34:   </head>
35:   <body onload="load()" onunload="GUnload()" style="height: 100%; margin: 0px;">
36:     <div id="map" style="height: 100%; width: 100%;"></div>
37:   </body>
38: </html>
39:
```

```
1: <!--
2:
3: map3.html
4:
5: Demonstrates map with controls.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12:
13: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml" style="height: 100%;">
17:   <head>
18:     <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA8igYd929VTmOEMLNjNyPlxQBpXogqfcSB7goB
POHTPn2IeSNTRQdD9mEVL94XCoVb0q6KdbS4ouIDA" type="text/javascript"></script>
19:     <script type="text/javascript">
20:       <![CDATA[
21:
22:         function load()
23:         {
24:             if (GBrowserIsCompatible())
25:             {
26:                 // instantiate map
27:                 var map = new GMap2(document.getElementById("map"));
28:
29:                 // center map on Science Center
30:                 map.setCenter(new GLatLng(42.376649, -71.115789), 13);
31:
32:                 // add control using a local variable
33:                 var typeControl = new GMapTypeControl();
34:                 map.addControl(typeControl);
35:
36:                 // add another control without using a local variable
37:                 map.addControl(new GLargeMapControl());
38:
39:                 // enable scroll wheel and smooth zooming
40:                 map.enableScrollWheelZoom();
41:                 map.enableContinuousZoom();
42:             }
43:         }
44:
45:       </script>
46:     </script>
```

```
47:     <title></title>
48: </head>
49: <body onload="load()" onunload="GUnload()" style="height: 100%; margin: 0px;">
50:     <div id="map" style="height: 100%; width: 100%;"></div>
51: </body>
52: </html>
53:
```

```
1: <!--
2:
3: map4.html
4:
5: Demonstrates map with marker and info window.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12:
13: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml" style="height: 100%;">
17:   <head>
18:     <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA8igYd929VTmOEMLNjNyP1xQBpXogqfcSB7goB
POHTPn2IeSNTRQdD9mEVL94XCoVb0q6KdbS4ouIDA" type="text/javascript"></script>
19:     <script type="text/javascript">
20:       /*
21:
22:         function load()
23:         {
24:             if (GBrowserIsCompatible())
25:             {
26:                 // instantiate map
27:                 var map = new GMap2(document.getElementById("map"));
28:
29:                 // prepare point
30:                 var point = new GLatLng(42.376649, -71.115789);
31:
32:                 // center map on Science Center
33:                 map.setCenter(point, 13);
34:
35:                 // mark Science Center
36:                 var marker = new GMarker(point);
37:                 map.addOverlay(marker);
38:
39:                 // associate info window with marker
40:                 GEvent.addListener(marker, "click", function() {
41:
42:                     // prepare XHTML
43:                     var xhtml = "&lt;b&gt;Science Center&lt;/b&gt;";
44:                     xhtml += "&lt;br /&gt;&lt;br /&gt;";
45:                     xhtml += "&lt;a href='http://en.wikipedia.org/wiki/Harvard_Science_Center' target='_blank'&gt;";
46:                     xhtml += "http://en.wikipedia.org/wiki/Harvard_Science_Center";</pre></div>
```



```
47:             xhtml += "</a>";
48:
49:             // open info window
50:             map.openInfoWindowHtml(point, xhtml);
51:
52:         });
53:
54:     }
55: }
56:
57: //]]>
58: </script>
59: <title></title>
60: </head>
61: <body onload="load()" onunload="GUnload()" style="height: 100%; margin: 0px;">
62:   <div id="map" style="height: 100%; width: 100%;"></div>
63: </body>
64: </html>
65:
```

```
1: <!--
2:
3: map5.html
4:
5: Demonstrates map with marker and info window.
6:
7: Also demonstrates Ajax via Google's GXmlHttp factory.
8:
9: Computer Science 50
10: David J. Malan
11:
12: -->
13:
14:
15: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
16:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
17:
18: <html xmlns="http://www.w3.org/1999/xhtml" style="height: 100%;">
19:   <head>
20:     <script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA8igYd929VTmOEMLNjNyP1xQBpXogqfcSB7goB
POHTPn2IeSNTRQdD9mEVL94XCoVb0q6KdbS4ouIDA" type="text/javascript"></script>
21:     <script type="text/javascript">
22:       <![CDATA[
23:
24:         function load()
25:         {
26:           if (GBrowserIsCompatible())
27:           {
28:             // instantiate map
29:             var map = new GMap2(document.getElementById("map"));
30:
31:             // prepare point
32:             var point = new GLatLng(37.4419, -122.1419);
33:
34:             // center map on Science Center
35:             map.setCenter(point, 13);
36:
37:             // mark Science Center
38:             var marker = new GMarker(point);
39:             map.addOverlay(marker);
40:
41:             // associate info window with marker
42:             GEvent.addListener(marker, "click", function() {
43:
44:               // prepare Ajax call
45:               var request = GXmlHttp.create();
46:               request.open("GET", "quote3.php?symbols=GOOG", true);
```

```
47:         request.onreadystatechange = function() {
48:
49:             // only handle successful calls
50:             if (request.readyState == 4 && request.status == 200)
51:             {
52:                 // prepare XHTML
53:                 var xhtml = "<b>Google</b>";
54:                 xhtml += "<br /><br />";
55:
56:                 // get Google's quote
57:                 var quote = request.responseXML.getElementsByTagName("quote")[0];
58:
59:                 // get Google's price
60:                 var price = quote.getElementsByTagName("price")[0].firstChild.nodeValue;
61:
62:                 // report price
63:                 xhtml += "$" + price;
64:
65:                 // open info window
66:                 map.openInfoWindowHtml(point, xhtml);
67:             }
68:         };
69:
70:         // get quote
71:         request.send(null);
72:
73:     });
74:
75: }
76:
77:
78: </script>
79: </script>
80: <title></title>
81: </head>
82: <body onload="load()" onunload="GUnload()" style="height: 100%; margin: 0px;">
83:   <div id="map" style="height: 100%; width: 100%;"></div>
84: </body>
85: </html>
86:
```

```
1: <?php
2:
3: /**
4:  * quote3.php
5:  *
6:  * Outputs price, low, and high of given symbols as XML.
7:  *
8:  * Computer Science 50
9:  * David J. Malan
10:  */
11:
12:
13: // massage input into CSV
14: $symbols = str_replace(", ", " ", $_GET["symbols"]);
15: $symbols = strtoupper($symbols);
16: $symbols = preg_split("/\s+/", $symbols);
17: $symbols = implode(" ", $symbols);
18:
19: // prepare to output results
20: header("Content-type: text/xml");
21: print("<quotes>\n");
22:
23: // get quotes
24: $fp = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s=$symbols&f=snlhg", "r");
25: while (($data = fgetcsv($fp)) !== FALSE)
26: {
27:     print(" <quote symbol='{ $data[0] }' name=' " . htmlentities($data[1]) . "'>\n");
28:     print("   <price>{ $data[2] }</price>\n");
29:     print("   <high>{ $data[3] }</high>\n");
30:     print("   <low>{ $data[4] }</low>\n");
31:     print(" </quote>\n");
32: }
33: fclose($fp);
34:
35: // that's all folks
36: print("</quotes>");
37: ?>
```