

```
1: <!--
2:
3: ajax1.html
4:
5: Gets stock quote from quotel.php via Ajax, displaying result with alert().
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script type="text/javascript">
19:       // 
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:           }
</pre></div><div data-bbox="127 527 179 550" data-label="Page-Header">11/17/08<br/>09:08:21</div><div data-bbox="389 539 607 553" data-label="Page-Header">lectures/weeks/9/src/ajax/ajax1.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 690 915" data-label="Text"><pre>48:
49:         // construct URL
50:         var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
51:
52:         // get quote
53:         xhr.onreadystatechange = handler;
54:         xhr.open("GET", url, true);
55:         xhr.send(null);
56:     }
57:
58:
59:     /*
60:      * void
61:      * handler()
62:      *
63:      * Handles the Ajax response.
64:      */
65:     function handler()
66:     {
67:       // only handle loaded requests
68:       if (xhr.readyState == 4)
69:       {
70:         if (xhr.status == 200)
71:           alert(xhr.responseText);
72:         else
73:           alert("Error with Ajax call!");
74:       }
75:     }
76:
77:     // ]]&gt;
78:   &lt;/script&gt;
79:   &lt;title&gt;&lt;/title&gt;
80: &lt;/head&gt;
81: &lt;body&gt;
82:   &lt;form onsubmit="quote(); return false;"&gt;
83:     Symbol: &lt;input id="symbol" type="text" /&gt;
84:     &lt;br /&gt;&lt;br /&gt;
85:     &lt;input type="submit" value="Get Quote" /&gt;
86:   &lt;/form&gt;
87: &lt;/body&gt;
88: &lt;/html&gt;
</pre></div>
```

```
1: <!--
2:
3: ajax2.html
4:
5: Gets stock quote from quotel.php via Ajax, embedding result in page itself.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script type="text/javascript">
19:       // 
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:           }
</pre></div><div data-bbox="127 527 179 550" data-label="Page-Header">11/17/08<br/>09:08:41</div><div data-bbox="389 539 607 553" data-label="Page-Header">lectures/weeks/9/src/ajax/ajax2.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 690 933" data-label="Text"><pre>48:
49:         // construct URL
50:         var url = "quotel.php?symbol=" + document.getElementById("symbol").value;
51:
52:         // get quote
53:         xhr.onreadystatechange = handler;
54:         xhr.open("GET", url, true);
55:         xhr.send(null);
56:     }
57:
58:
59:     /*
60:      * void
61:      * handler()
62:      *
63:      * Handles the Ajax response.
64:      */
65:     function handler()
66:     {
67:       // only handle loaded requests
68:       if (xhr.readyState == 4)
69:       {
70:         if (xhr.status == 200)
71:           document.getElementById("price").innerHTML = xhr.responseText;
72:         else
73:           alert("Error with Ajax call!");
74:       }
75:     }
76:
77: // ]]&gt;
78: &lt;/script&gt;
79: &lt;title&gt;&lt;/title&gt;
80: &lt;/head&gt;
81: &lt;body&gt;
82:   &lt;form onsubmit="quote(); return false;"&gt;
83:     Symbol: &lt;input id="symbol" type="text" /&gt;
84:     &lt;br /&gt;
85:     Price: &lt;span id="price"&gt;&lt;b&gt;to be determined&lt;/b&gt;&lt;/span&gt;
86:     &lt;br /&gt;&lt;br /&gt;
87:     &lt;input type="submit" value="Get Quote" /&gt;
88:   &lt;/form&gt;
89: &lt;/body&gt;
90: &lt;/html&gt;
</pre></div>
```

```
1: <!--
2:
3: ajax3.html
4:
5: Gets stock quote (plus day's low and high) from quote2.php via Ajax,
6: embedding result in page itself after indicating progress with an
7: animated GIF.
8:
9: Computer Science 50
10: David J. Malan
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><div data-bbox="127 527 179 550" data-label="Page-Header">11/17/08<br/>09:09:22</div><div data-bbox="389 539 607 553" data-label="Page-Header">lectures/weeks/9/src/ajax/ajax3.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 690 968" data-label="Text"><pre>48:             return;
49:           }
50:
51:           // construct URL
52:           var url = "quote2.php?symbol=" + document.getElementById("symbol").value;
53:
54:           // show progress
55:           document.getElementById("progress").style.display = "block";
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:             // 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:       // ]]&gt;
86:     &lt;/script&gt;
87:   &lt;title&gt;&lt;/title&gt;
88: &lt;/head&gt;
89: &lt;body&gt;
90:   &lt;form onsubmit="quote(); return false;"&gt;
91:     Symbol: &lt;input id="symbol" type="text" /&gt;
92:     &lt;br /&gt;&lt;br /&gt;
93:     &lt;div id="progress" style="display: none;"&gt;
94:       &lt;img alt="Please Wait" src="pleasewait.gif" /&gt;</pre></div>
```

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

```
1: <?php
2:
3:   /**
4:    * quote1.php
5:    *
6:    * Outputs price of given symbol as text/html.
7:    *
8:    * Computer Science 50
9:    * David J. Malan
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: <?php
2:
3:     /**
4:      * quote2.php
5:      *
6:      * Outputs price, low, and high of given symbol as text/html, after
7:      * inserting an artificial delay.
8:      *
9:      * Computer Science 50
10:     * David J. Malan
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: form1.html
4:
5: A form without client-side validation.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:     "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:     <head>
18:         <title></title>
19:     </head>
20:     <body>
21:         <form action="process.php" method="get">
22:             Email: <input name="email" type="text" />
23:             <br />
24:             Password: <input name="password1" type="password" />
25:             <br />
26:             Password (again): <input name="password2" type="password" />
27:             <br />
28:             I agree to the terms and conditions: <input name="agreement" type="checkbox" />
29:             <br /><br />
30:             <input type="submit" value="Submit" />
31:         </form>
32:     </body>
33: </html>
```

```
1: <!--
2:
3: form2.html
4:
5: A form with client-side validation.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script type="text/javascript">
19:       // 
20:
21:         function validate()
22:         {
23:             if (document.forms.registration.email.value == "")
24:             {
25:                 alert("You must provide an email address.");
26:                 return false;
27:             }
28:             else if (document.forms.registration.password1.value == "")
29:             {
30:                 alert("You must provide a password.");
31:                 return false;
32:             }
33:             else if (document.forms.registration.password1.value != document.forms.registration.password2.value)
34:             {
35:                 alert("You must provide the same password twice.");
36:                 return false;
37:             }
38:             else if (!document.forms.registration.agreement.checked)
39:             {
40:                 alert("You must agree to our terms and conditions.");
41:                 return false;
42:             }
43:             return true;
44:         }
45:
46:       // ]]&gt;
47:     &lt;/script&gt;</pre></div><div data-bbox="126 527 179 550" data-label="Page-Header">11/17/08<br/>09:12:13</div><div data-bbox="381 539 613 553" data-label="Page-Header">lectures/weeks/9/src/forms/form2.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 748 697" data-label="Text"><pre>48:     &lt;title&gt;&lt;/title&gt;
49:   &lt;/head&gt;
50:   &lt;body&gt;
51:     &lt;form action="process.php" method="get" name="registration" onsubmit="return validate();" &gt;
52:       Email: &lt;input name="email" type="text" /&gt;
53:       &lt;br /&gt;
54:       Password: &lt;input name="password1" type="password" /&gt;
55:       &lt;br /&gt;
56:       Password (again): &lt;input name="password2" type="password" /&gt;
57:       &lt;br /&gt;
58:       I agree to the terms and conditions: &lt;input name="agreement" type="checkbox" /&gt;
59:       &lt;br /&gt;&lt;br /&gt;
60:       &lt;input type="submit" value="Submit" /&gt;
61:     &lt;/form&gt;
62:   &lt;/body&gt;
63: &lt;/html&gt;</pre></div>
```

```
1: <!--
2:
3: form3.html
4:
5: A form with client-side validation demonstrating "this" keyword.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script type="text/javascript">
19:       // 
20:
21:         function validate(f)
22:         {
23:           if (f.email.value == "")
24:           {
25:             alert("You must provide an email address.");
26:             return false;
27:           }
28:           else if (f.password1.value == "")
29:           {
30:             alert("You must provide a password.");
31:             return false;
32:           }
33:           else if (f.password1.value != f.password2.value)
34:           {
35:             alert("You must provide the same password twice.");
36:             return false;
37:           }
38:           else if (!f.agreement.checked)
39:           {
40:             alert("You must agree to our terms and conditions.");
41:             return false;
42:           }
43:           return true;
44:         }
45:
46:       // ]&gt;
47:     &lt;/script&gt;</pre></div><div data-bbox="127 527 180 550" data-label="Page-Header">11/17/08<br/>09:12:47</div><div data-bbox="381 538 613 553" data-label="Page-Header">lectures/weeks/9/src/forms/form3.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 555 773 697" data-label="Text"><pre>48:     &lt;title&gt;&lt;/title&gt;
49:   &lt;/head&gt;
50:   &lt;body&gt;
51:     &lt;form action="process.php" method="get" name="registration" onsubmit="return validate(this);"&gt;
52:       Email: &lt;input name="email" type="text" /&gt;
53:       &lt;br /&gt;
54:       Password: &lt;input name="password1" type="password" /&gt;
55:       &lt;br /&gt;
56:       Password (again): &lt;input name="password2" type="password" /&gt;
57:       &lt;br /&gt;
58:       I agree to the terms and conditions: &lt;input name="agreement" type="checkbox" /&gt;
59:       &lt;br /&gt;&lt;br /&gt;
60:       &lt;input type="submit" value="Submit" /&gt;
61:     &lt;/form&gt;
62:   &lt;/body&gt;
63: &lt;/html&gt;</pre></div>
```

```
1: <!--
2:
3: form4.html
4:
5: A form with client-side validation demonstrating disabled property.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script type="text/javascript">
19:       // 
20:
21:         function toggle()
22:         {
23:           if (document.forms.registration.button.disabled)
24:             document.forms.registration.button.disabled = false;
25:           else
26:             document.forms.registration.button.disabled = true;
27:         }
28:
29:         function validate()
30:         {
31:           if (document.forms.registration.email.value == "")
32:           {
33:             alert("You must provide an email address.");
34:             return false;
35:           }
36:           else if (document.forms.registration.password1.value == "")
37:           {
38:             alert("You must provide a password.");
39:             return false;
40:           }
41:           else if (document.forms.registration.password1.value != document.forms.registration.password2.value)
42:           {
43:             alert("You must provide the same password twice.");
44:             return false;
45:           }
46:           else if (!document.forms.registration.agreement.checked)
47:           {</pre></div><div data-bbox="127 527 179 550" data-label="Page-Header">11/17/08<br/>09:12:42</div><div data-bbox="381 539 613 552" data-label="Page-Header">lectures/weeks/9/src/forms/form4.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 556 816 767" data-label="Text"><pre>48:             alert("You must agree to our terms and conditions.");
49:             return false;
50:           }
51:           return true;
52:         }
53:
54:       // ]]&gt;
55:     &lt;/script&gt;
56:     &lt;title&gt;&lt;/title&gt;
57:   &lt;/head&gt;
58:   &lt;body&gt;
59:     &lt;form action="process.php" method="get" name="registration" onsubmit="return validate();"&gt;
60:       Email: &lt;input name="email" type="text" /&gt;
61:       &lt;br /&gt;
62:       Password: &lt;input name="password1" type="password" /&gt;
63:       &lt;br /&gt;
64:       Password (again): &lt;input name="password2" type="password" /&gt;
65:       &lt;br /&gt;
66:       I agree to the terms and conditions: &lt;input name="agreement" onclick="toggle();" type="checkbox" /&gt;
67:       &lt;br /&gt;&lt;br /&gt;
68:       &lt;input disabled="disabled" name="button" type="submit" value="Submit" /&gt;
69:     &lt;/form&gt;
70:   &lt;/body&gt;
71: &lt;/html&gt;</pre></div>
```



```
1: <!--
2:
3: form5.html
4:
5: A form with client-side validation demonstrating regular expressions.
6:
7: Computer Science 50
8: David J. Malan
9:
10: -->
11:
12: <!DOCTYPE html PUBLIC
13:   "-//W3C//DTD XHTML 1.0 Transitional//EN"
14:   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15:
16: <html xmlns="http://www.w3.org/1999/xhtml">
17:   <head>
18:     <script type="text/javascript">
19:       // 
20:
21:         function validate()
22:         {
23:             if (!document.forms.registration.email.value.match(/.+@.+\.edu$/))
24:             {
25:                 alert("You must provide a .edu email address.");
26:                 return false;
27:             }
28:             else if (document.forms.registration.password1.value == "")
29:             {
30:                 alert("You must provide a password.");
31:                 return false;
32:             }
33:             else if (document.forms.registration.password1.value != document.forms.registration.password2.value)
34:             {
35:                 alert("You must provide the same password twice.");
36:                 return false;
37:             }
38:             else if (!document.forms.registration.agreement.checked)
39:             {
40:                 alert("You must agree to our terms and conditions.");
41:                 return false;
42:             }
43:             return true;
44:         }
45:
46:       // ]&gt;
47:     &lt;/script&gt;</pre></div><div data-bbox="126 527 180 550" data-label="Page-Header">11/17/08<br/>09:12:39</div><div data-bbox="381 538 613 552" data-label="Page-Header">lectures/weeks/9/src/forms/form5.html</div><div data-bbox="833 529 853 551" data-label="Page-Header">2</div><div data-bbox="137 555 748 697" data-label="Text"><pre>48:     &lt;title&gt;&lt;/title&gt;
49:   &lt;/head&gt;
50:   &lt;body&gt;
51:     &lt;form action="process.php" method="get" name="registration" onsubmit="return validate();"&gt;
52:       Email: &lt;input name="email" type="text" /&gt;
53:       &lt;br /&gt;
54:       Password: &lt;input name="password1" type="password" /&gt;
55:       &lt;br /&gt;
56:       Password (again): &lt;input name="password2" type="password" /&gt;
57:       &lt;br /&gt;
58:       I agree to the terms and conditions: &lt;input name="agreement" type="checkbox" /&gt;
59:       &lt;br /&gt;&lt;br /&gt;
60:       &lt;input type="submit" value="Submit" /&gt;
61:     &lt;/form&gt;
62:   &lt;/body&gt;
63: &lt;/html&gt;</pre></div>
```

```
1: /**
2:  * process.php
3:  *
4:  * Dumps HTTP requests (albeit as invalid XHTML).
5:  *
6:  * Computer Science 50
7:  * David J. Malan
8:  */
9:
10: <pre>
11:  <?php print_r($_REQUEST); ?>
12: </pre>
13:
```

```
1: <?php
2:
3:  /*****
4:   * dictionary.php
5:   *
6:   * Computer Science 50
7:   * David J. Malan
8:   *
9:   * Implements a dictionary.
10:  *****/
11:
12:
13:  // size of dictionary
14:  $size = 0;
15:
16:  // dictionary
17:  $dictionary = array();
18:
19:
20:  /*
21:   * bool
22:   * check($word)
23:   *
24:   * Returns TRUE if word is in dictionary else FALSE.
25:  */
26:
27:  function check($word)
28:  {
29:      global $dictionary;
30:      if ($dictionary[strtolower($word)])
31:          return TRUE;
32:      else
33:          return FALSE;
34:  }
35:
36:
37:
38:  /*
39:   * bool
40:   * load($dict)
41:   *
42:   * Loads dict into memory. Returns TRUE if successful else FALSE.
43:  */
44:
45:  function load($dict)
46:  {
47:      global $dictionary, $size;
```

```
48:         if (($fp = fopen($dict, "r")) === FALSE)
49:             return FALSE;
50:         while ((fscanf($fp, "%s", $word) == 1)
51:             {
52:             $dictionary[$word] = TRUE;
53:             $size++;
54:             }
55:         fclose($fp);
56:         return TRUE;
57:     }
58:
59:
60:     /*
61:     * int
62:     * size()
63:     *
64:     * Returns number of words in dictionary if loaded else 0 if not yet loaded.
65:     */
66:
67:     function size()
68:     {
69:         global $size;
70:         return $size;
71:     }
72:
73:
74:     /*
75:     * int
76:     * unload()
77:     *
78:     * Unloads dictionary from memory. Returns TRUE if successful else FALSE.
79:     */
80:
81:     function unload()
82:     {
83:         return TRUE;
84:     }
85: ?>
```

```
1: #!/usr/local/bin/php
2: <?php
3:
4:     /*****
5:     * speller.php
6:     *
7:     * Computer Science 50
8:     * David J. Malan
9:     *
10:    * Implements a spell-checker.
11:    *****/
12:
13:     require("dictionary.php");
14:
15:     // suppress notices and warnings
16:     error_reporting(E_ALL ^ E_NOTICE ^ E_WARNING);
17:
18:
19:     // maximum length for a word
20:     // (e.g., pneumonoultramicroscopicsilicovolcanoconiosis)
21:     define("LENGTH", 45);
22:
23:     // default dictionary
24:     define("WORDS", "/home/cs50/pub/share/pset6/dict/words");
25:
26:     // check for correct number of args
27:     if ($argc != 2 && $argc != 3)
28:     {
29:         print("Usage: speller.php [dict] file\n");
30:         return 1;
31:     }
32:
33:     // benchmarks
34:     $ti_load = 0.; $ti_check = 0.; $ti_size = 0.; $ti_unload = 0.;
35:
36:     // determine dictionary to use
37:     $dict = ($argc == 3) ? $argv[1] : WORDS;
38:
39:     // load dictionary
40:     $before = microtime(TRUE);
41:     $loaded = load($dict);
42:     $after = microtime(TRUE);
43:
44:     // abort if dictionary not loaded
45:     if (!$loaded)
46:     {
47:         print("Could not load $dict.\n");
```

```
48:         return 2;
49:     }
50:
51:     // calculate time to load dictionary
52:     $ti_load = $after - $before;
53:
54:     // try to open file
55:     $file = ($argc == 3) ? $argv[2] : $argv[1];
56:     $fp = fopen($file, "r");
57:     if ($fp === FALSE)
58:     {
59:         print("Could not open $file.\n");
60:         return 3;
61:     }
62:
63:     // prepare to report misspellings
64:     printf("\nMISPELLED WORDS\n\n");
65:
66:     // prepare to spell-check
67:     $word = "";
68:     $index = 0; $mispellings = 0; $words = 0;
69:
70:     // spell-check each word in file
71:     for ($c = fgetc($fp); $c != FALSE; $c = fgetc($fp))
72:     {
73:         // allow alphabetical characters and apostrophes (for possessives)
74:         if (preg_match("/[a-zA-Z]/", $c) || ($c == "'" && $index > 0))
75:         {
76:             // append character to word
77:             $word .= $c;
78:             $index++;
79:
80:             // ignore alphabetical strings too long to be words
81:             if ($index >= LENGTH)
82:             {
83:                 // consume remainder of alphabetical string
84:                 while (($c = fgetc($fp)) != FALSE && preg_match("/[a-zA-Z]/", $c));
85:
86:                 // prepare for new word
87:                 $index = 0; $word = "";
88:             }
89:         }
90:
91:         // ignore words with numbers (like MS Word)
92:         else if (ctype_digit($c))
93:         {
94:             // consume remainder of alphabetical string
```

```
95:         while (($c = fgetc($fp)) != FALSE && preg_match("/[a-zA-z0-9]", $c));
96:
97:         // prepare for new word
98:         $index = 0; $word = "";
99:     }
100:
101:     // we must have found a whole word
102:     else if ($index > 0)
103:     {
104:         // update counter
105:         $words++;
106:
107:         // check word's spelling
108:         $before = microtime(TRUE);
109:         $misspelled = !check($word);
110:         $after = microtime(TRUE);
111:
112:         // update benchmark
113:         $ti_check += $after - $before;
114:
115:         // print word if misspelled
116:         if ($misspelled)
117:         {
118:             print("$word\n");
119:             $mispellings++;
120:         }
121:
122:         // prepare for next word
123:         $index = 0; $word = "";
124:     }
125: }
126:
127: // close file
128: fclose($fp);
129:
130: // determine dictionary's size
131: $before = microtime(TRUE);
132: $n = size();
133: $after = microtime(TRUE);
134:
135: // calculate time to determine dictionary's size
136: $ti_size = $after - $before;
137:
138: // unload dictionary
139: $before = microtime(TRUE);
140: $unloaded = unload();
141: $after = microtime(TRUE);
```

```
142:
143: // abort if dictionary not unloaded
144: if (!$unloaded)
145: {
146:     print("Could not load $dict.\n");
147:     return 5;
148: }
149: // calculate time to determine dictionary's size
150: $ti_unload = $after - $before;
151:
152: // report benchmarks
153: printf("\nWORDS MISPELLED:      %d\n", $mispellings);
154: printf("WORDS IN DICTIONARY:  %d\n", $n);
155: printf("WORDS IN FILE:        %d\n", $words);
156: printf("TIME IN load:         %f\n", $ti_load);
157: printf("TIME IN check:        %f\n", $ti_check);
158: printf("TIME IN size:         %f\n", $ti_size);
159: printf("TIME IN unload:       %f\n", $ti_unload);
160: printf("TOTAL TIME:          %f\n", $ti_load + $ti_check + $ti_size + $ti_unload);
161:
162: ?>
```