```
<!--

ajax1.html

Gets stock quote from quote1.php via Ajax, displaying result with alert().

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        // an XMLHttpRequest
        var xhr = null;

        /*
         * void
         * quote()
         *
         * Gets a quote.
         */
        function quote()
        {
            // instantiate XMLHttpRequest object
            try
            {
                xhr = new XMLHttpRequest();
            }
            catch (e)
            {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }

            // handle old browsers
            if (xhr == null)
            {
                alert("Ajax not supported by your browser!");
                return;
            }
```



```

            // construct URL
            var url = "quote1.php?symbol=" + document.getElementById("symbol").value;

            // get quote
            xhr.onreadystatechange = handler;
            xhr.open("GET", url, true);
            xhr.send(null);
        }


        /*
         * void
         * handler()
         *
         * Handles the Ajax response.
         */
        function handler()
        {
            // only handle loaded requests
            if (xhr.readyState == 4)
            {
                // display response if possible
                if (xhr.status == 200)
                    alert(xhr.responseText);
                else
                    alert("Error with Ajax call!");
            }
        }

    // ]]>
    </script>
    <title>ajax1</title>
  </head>
  <body>
    <form onsubmit="quote(); return false;">
      Symbol: <input id="symbol" type="text">
      <br><br>
      <input type="submit" value="Get Quote">
    </form>
  </body>
</html>
```

```
<!--

ajax2.html

Gets stock quote from quote1.php via Ajax, embedding result in page itself.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        // an XMLHttpRequest
        var xhr = null;

        /*
         * void
         * quote()
         *
         * Gets a quote.
         */
        function quote()
        {
            // instantiate XMLHttpRequest object
            try
            {
                xhr = new XMLHttpRequest();
            }
            catch (e)
            {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }

            // handle old browsers
            if (xhr == null)
            {
                alert("Ajax not supported by your browser!");
                return;
            }
```



```

            // construct URL
            var url = "quote1.php?symbol=" + document.getElementById("symbol").value;

            // get quote
            xhr.onreadystatechange = handler;
            xhr.open("GET", url, true);
            xhr.send(null);
        }


        /*
         * void
         * handler()
         *
         * Handles the Ajax response.
         */
        function handler()
        {
            // only handle loaded requests
            if (xhr.readyState == 4)
            {
                // embed respose in page if possible
                if (xhr.status == 200)
                    document.getElementById("price").innerHTML = xhr.responseText;
                else
                    alert("Error with Ajax call!");
            }
        }

    // ]]>
    </script>
    <title>ajax2</title>
  </head>
  <body>
    <form onsubmit="quote(); return false;">
      Symbol: <input id="symbol" type="text">
      <br>
      Price: <span id="price"><b>to be determined</b></span>
      <br><br>
      <input type="submit" value="Get Quote">
    </form>
  </body>
</html>
```

```
<!--

ajax3.html

Gets stock quote (plus day's low and high) from quote2.php via Ajax,
embedding result in page itself after indicating progress with an
animated GIF.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        // an XMLHttpRequest
        var xhr = null;

        /*
         * void
         * quote()
         *
         * Gets a quote.
         */
        function quote()
        {
            // instantiate XMLHttpRequest object
            try
            {
                xhr = new XMLHttpRequest();
            }
            catch (e)
            {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }

            // handle old browsers
            if (xhr == null)
            {
                alert("Ajax not supported by your browser!");
```



```
                return;
            }

            // construct URL
            var url = "quote2.php?symbol=" + document.getElementById("symbol").value;

            // show progress
            document.getElementById("progress").style.display = "block";

            // get quote
            xhr.onreadystatechange = handler;
            xhr.open("GET", url, true);
            xhr.send(null);
        }


        /*
         * void
         * handler()
         *
         * Handles the Ajax response.
         */
        function handler()
        {
            // only handle requests in "loaded" state
            if (xhr.readyState == 4)
            {
                // hide progress
                document.getElementById("progress").style.display = "none";

                // embed response in page if possible
                if (xhr.status == 200)
                    document.getElementById("quote").innerHTML = xhr.responseText;
                else
                    alert("Error with Ajax call!");
            }
        }

    // ]]>
    </script>
    <title>ajax3</title>
  </head>
  <body>
    <form onsubmit="quote(); return false;">
      Symbol: <input id="symbol" type="text">
```

```
      <br><br>
      <div id="progress" style="display: none">
        <img alt="Please Wait" src="pleasewait.gif">
        <br><br>
      </div>
      <div id="quote"></div>
      <br><br>
      <input type="submit" value="Get Quote">
    </form>
  </body>
</html>
```



```
<!--

ajax4.html

Gets stock quote from quote1.php via Ajax, displaying result with alert().
Implements handler as an anonymous function.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        // an XMLHttpRequest
        var xhr = null;

        /*
         * void
         * quote()
         *
         * Gets a quote.
         */
        function quote()
        {
            // instantiate XMLHttpRequest object
            try
            {
                xhr = new XMLHttpRequest();
            }
            catch (e)
            {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }

            // handle old browsers
            if (xhr == null)
            {
                alert("Ajax not supported by your browser!");
                return;
```

```
            }

            // construct URL
            var url = "quote1.php?symbol=" + document.getElementById("symbol").value;

            // get quote
            xhr.onreadystatechange = function () {
                // only handle loaded requests
                if (xhr.readyState == 4)
                {
                    // display response if possible
                    if (xhr.status == 200)
                        alert(xhr.responseText);
                    else
                        alert("Error with Ajax call!");
                }
            };
            xhr.open("GET", url, true);
            xhr.send(null);
        }

    // ]]>
    </script>
    <title>ajax4</title>
  </head>
  <body>
    <form onsubmit="quote(); return false;">
      Symbol: <input id="symbol" type="text">
      <br><br>
      <input type="submit" value="Get Quote">
    </form>
  </body>
</html>
```



```
<!--

ajax5.html

Gets stock quote (plus day's low and high) from quote3.php via Ajax,
embedding (JSON) result in page itself.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        // an XMLHttpRequest
        var xhr = null;

        /*
         * void
         * quote()
         *
         * Gets a quote.
         */
        function quote()
        {
            // instantiate XMLHttpRequest object
            try
            {
                xhr = new XMLHttpRequest();
            }
            catch (e)
            {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }

            // handle old browsers
            if (xhr == null)
            {
                alert("Ajax not supported by your browser!");
                return;
```

```
            }

            // construct URL
            var url = "quote3.php?symbol=" + document.getElementById("symbol").value;

            // get quote
            xhr.onreadystatechange = function() {

                // only handle requests in "loaded" state
                if (xhr.readyState == 4)
                {
                    // embed response in page if possible
                    if (xhr.status == 200)
                    {
                        var quote = eval("(" + xhr.responseText + ")");
                        document.getElementById("price").innerHTML = quote.price;
                        document.getElementById("high").innerHTML = quote.high;
                        document.getElementById("low").innerHTML = quote.low;
                    }
                    else
                        alert("Error with Ajax call!");
                }

            };
            xhr.open("GET", url, true);
            xhr.send(null);
        }


        /*
         * void
         * handler()
         *
         * Handles the Ajax response.
         */
        function handler()
        {
        }

    // ]]>
    </script>
    <title>ajax5</title>
  </head>
  <body>
    <form onsubmit="quote(); return false;">
```



```
      Symbol: <input id="symbol" type="text">
      <br><br>
      Price: <span id="price"></span>
      <br>
      High: <span id="high"></span>
      <br>
      Low: <span id="low"></span>
      <br>
      <br><br>
      <input type="submit" value="Get Quote">
    </form>
  </body>
</html>
```

```
<?

    /**
     * quote1.php
     *
     * Outputs price of given symbol as text/html.
     *
     * Computer Science 50
     * David J. Malan
     */

    // get quote
    $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET["symbol"]}&f=ell1", "r");
    if ($handle !== FALSE)
    {
        $data = fgetcsv($handle);
        if ($data !== FALSE && $data[0] == "N/A")
            print($data[1]);
        fclose($handle);
    }
?>
```



```
<?

    /**
     * quote2.php
     *
     * Outputs price, low, and high of given symbol as text/html, after
     * inserting an artificial delay.
     *
     * Computer Science 50
     * David J. Malan
     */


    // pretend server is slow
    sleep(5);

    // try to get quote
    $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET["symbol"]}&f=ell1hg", "r");
    if ($handle !== FALSE)
    {
        $data = fgetcsv($handle);
        if ($data !== FALSE && $data[0] == "N/A")
        {
            print("Price: {$data[1]}");
            print("<br>");
            print("High: {$data[2]}");
            print("<br>");
            print("Low: {$data[3]}");
        }
        fclose($handle);
    }
?>
```

```
<?

    /**
     * quote3.php
     *
     * Outputs price, low, and high of given symbol as JSON.
     *
     * Computer Science 50
     * David J. Malan
     */

    // try to get quote
    $quote = array();
    $handle = @fopen("http://download.finance.yahoo.com/d/quotes.csv?s={$_GET["symbol"]}&f=e1llhg", "r");
    if ($handle !== FALSE)
    {
        $data = fgetcsv($handle);
        if ($data !== FALSE && $data[0] == "N/A")
        {
            $quote["price"] = $data[1];
            $quote["high"] = $data[2];
            $quote["low"] = $data[3];
        }
        fclose($handle);
    }
    header("Content-type: application/json");
    print(json_encode($quote));
?>
```



```
<?

    /**
     * dump.php
     *
     * Dumps HTTP requests.
     *
     * Computer Science 50
     * David J. Malan
     */

?>

<!DOCTYPE html>

<html>
  <head>
    <title>dump</title>
  </head>
  <body>
    <pre><? print_r($_GET); ?></pre>
  </body>
</html>
```

```
<!--

form1.html

A form without client-side validation.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <title>form1</title>
  </head>
  <body>
    <form action="dump.php" method="get">
      Email: <input name="email" type="text">
      <br>
      Password: <input name="password1" type="password">
      <br>
      Password (again): <input name="password2" type="password">
      <br>
      I agree to the terms and conditions: <input name="agreement" type="checkbox">
      <br><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```



```
<!--

form2.html

A form with client-side validation.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        function validate()
        {
            if (document.forms.registration.email.value == "")
            {
                alert("You must provide an email adddress.");
                return false;
            }
            else if (document.forms.registration.password1.value == "")
            {
                alert("You must provide a password.");
                return false;
            }
            else if (document.forms.registration.password1.value != document.forms.registration.password2.value)
            {
                alert("You must provide the same password twice.");
                return false;
            }
            else if (!document.forms.registration.agreement.checked)
            {
                alert("You must agree to our terms and conditions.");
                return false;
            }
            return true;
        }

    // ]]>
    </script>
```

```
    <title>form2</title>
  </head>
  <body>
    <form action="dump.php" method="get" name="registration" onsubmit="return validate();">
      Email: <input name="email" type="text">
      <br>
      Password: <input name="password1" type="password">
      <br>
      Password (again): <input name="password2" type="password">
      <br>
      I agree to the terms and conditions: <input name="agreement" type="checkbox">
      <br><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```



```
<!--

form3.html

A form with client-side validation demonstrating "this" keyword.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        function validate(f)
        {
            if (f.email.value == "")
            {
                alert("You must provide an email adddress.");
                return false;
            }
            else if (f.password1.value == "")
            {
                alert("You must provide a password.");
                return false;
            }
            else if (f.password1.value != f.password2.value)
            {
                alert("You must provide the same password twice.");
                return false;
            }
            else if (!f.agreement.checked)
            {
                alert("You must agree to our terms and conditions.");
                return false;
            }
            return true;
        }

    // ]]>
    </script>
```

```
    <title>form3</title>
  </head>
  <body>
    <form action="dump.php" method="get" onsubmit="return validate(this);">
      Email: <input name="email" type="text">
      <br>
      Password: <input name="password1" type="password">
      <br>
      Password (again): <input name="password2" type="password">
      <br>
      I agree to the terms and conditions: <input name="agreement" type="checkbox">
      <br><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```



```
<!--

form4.html

A form with client-side validation demonstrating disabled property.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        function toggle()
        {
            if (document.forms.registration.button.disabled)
                document.forms.registration.button.disabled = false;
            else
                document.forms.registration.button.disabled = true;
        }

        function validate()
        {
            if (document.forms.registration.email.value == "")
            {
                alert("You must provide an email adddress.");
                return false;
            }
            else if (document.forms.registration.password1.value == "")
            {
                alert("You must provide a password.");
                return false;
            }
            else if (document.forms.registration.password1.value != document.forms.registration.password2.value)
            {
                alert("You must provide the same password twice.");
                return false;
            }
            else if (!document.forms.registration.agreement.checked)
            {
```

```
                alert("You must agree to our terms and conditions.");
                return false;
            }
            return true;
        }

    // ]]>
    </script>
    <title>form4</title>
  </head>
  <body>
    <form action="dump.php" method="get" name="registration" onsubmit="return validate();">
      Email: <input name="email" type="text">
      <br>
      Password: <input name="password1" type="password">
      <br>
      Password (again): <input name="password2" type="password">
      <br>
      I agree to the terms and conditions: <input name="agreement" onclick="toggle();" type="checkbox">
      <br><br>
      <input disabled="disabled" name="button" type="submit" value="Submit">
    </form>
  </body>
</html>
```



```
<!--

form5.html

A form with client-side validation demonstrating regular expressions.

Computer Science 50
David J. Malan

-->

<!DOCTYPE html>

<html>
  <head>
    <script type="text/javascript">
    // <![CDATA[

        function validate()
        {
            if (!document.forms.registration.email.value.match(/.+@.+\.edu$/))
            {
                alert("You must provide a .edu email adddress.");
                return false;
            }
            else if (document.forms.registration.password1.value == "")
            {
                alert("You must provide a password.");
                return false;
            }
            else if (document.forms.registration.password1.value != document.forms.registration.password2.value)
            {
                alert("You must provide the same password twice.");
                return false;
            }
            else if (!document.forms.registration.agreement.checked)
            {
                alert("You must agree to our terms and conditions.");
                return false;
            }
            return true;
        }

    // ]]>
    </script>
```

```
    <title>form5</title>
  </head>
  <body>
    <form action="dump.php" method="get" name="registration" onsubmit="return validate();">
      Email: <input name="email" type="text">
      <br>
      Password: <input name="password1" type="password">
      <br>
      Password (again): <input name="password2" type="password">
      <br>
      I agree to the terms and conditions: <input name="agreement" type="checkbox">
      <br><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```



```
<?

    /**********************************************************************
     * dictionary.php
     *
     * Computer Science 50
     * David J. Malan
     *
     * Implements a dictionary.
     **********************************************************************/


    // size of dictionary
    $size = 0;

    // dictionary
    $dictionary = array();


    /*
     * bool
     * check($word)
     *
     * Returns TRUE if word is in dictionary else FALSE.
     */

    function check($word)
    {
        global $dictionary;
        if ($dictionary[strtolower($word)])
            return TRUE;
        else
            return FALSE;
    }


    /*
     * bool
     * load($dict)
     *
     * Loads dict into memory.  Returns TRUE if successful else FALSE.
     */

    function load($dict)
    {
```

```
        global $dictionary, $size;
        if (!file_exists($dict) && is_readable($dict))
            return FALSE;
        foreach (file($dict) as $word)
        {
            $dictionary[chop($word)] = TRUE;
            $size++;
        }
        return TRUE;
    }


    /*
     * int
     * size()
     *
     * Returns number of words in dictionary if loaded else 0 if not yet loaded.
     */

    function size()
    {
        global $size;
        return $size;
    }


    /*
     * int
     * unload()
     *
     * Unloads dictionary from memory.  Returns TRUE if successful else FALSE.
     */

    function unload()
    {
        return TRUE;
    }

?>
```



```
#!/usr/bin/php
<?

    /**********************************************************************
     * speller.php
     *
     * Computer Science 50
     * David J. Malan
     *
     * Implements a spell-checker.
     **********************************************************************/

    require("dictionary.php");

    // suppress notices and warnings
    error_reporting(E_ALL ^ E_NOTICE);


    // maximum length for a word
    // (e.g., pneumonoultramicroscopicsilicovolcanoconiosis)
    define("LENGTH", 45);

    // default dictionary
    define("WORDS", "/home/cs50/pub/share/pset6/dict/words");

    // check for correct number of args
    if ($argc != 2 && $argc != 3)
    {
        print("Usage: speller.php [dict] file\n");
        return 1;
    }

    // benchmarks
    $ti_load = 0.0; $ti_check = 0.0; $ti_size = 0.0; $ti_unload = 0.0;

    // determine dictionary to use
    $dict = ($argc == 3) ? $argv[1] : WORDS;

    // load dictionary
    $before = microtime(TRUE);
    $loaded = load($dict);
    $after = microtime(TRUE);

    // abort if dictionary not loaded
    if (!$loaded)
```

```
    {
        print("Could not load $dict.\n");
        return 2;
    }

    // calculate time to load dictionary
    $ti_load = $after - $before;

    // try to open file
    $file = ($argc == 3) ? $argv[2] : $argv[1];
    $fp = fopen($file, "r");
    if ($fp === FALSE)
    {
        print("Could not open $file.\n");
        return 3;
    }

    // prepare to report misspellings
    printf("\nMISSPELLED WORDS\n\n");

    // prepare to spell-check
    $word  = "";
    $index = 0; $misspellings = 0; $words = 0;

    // spell-check each word in file
    for ($c = fgetc($fp); $c !== FALSE; $c = fgetc($fp))
    {
        // allow alphabetical characters and apostrophes (for possessives)
        if (preg_match("/[a-zA-Z]/", $c) || ($c == "'" && $index > 0))
        {
            // append character to word
            $word .= $c;
            $index++;

            // ignore alphabetical strings too long to be words
            if ($index >= LENGTH)
            {
                // consume remainder of alphabetical string
                while (($c = fgetc($fp)) !== FALSE && preg_match("/[a-zA-Z]/", $c));

                // prepare for new word
                $index = 0; $word = "";
            }
        }

```



```
        // ignore words with numbers (like MS Word)
        else if (ctype_digit($c))
        {
            // consume remainder of alphabetical string
            while (($c = fgetc($fp)) !== FALSE && preg_match("/[a-zA-z0-9]/", $c));

            // prepare for new word
            $index = 0; $word = "";
        }

        // we must have found a whole word
        else if ($index > 0)
        {
            // update counter
            $words++;

            // check word's spelling
            $before = microtime(TRUE);
            $misspelled = !check($word);
            $after = microtime(TRUE);

            // update benchmark
            $ti_check += $after - $before;

            // print word if misspelled
            if ($misspelled)
            {
                print("$word\n");
                $misspellings++;
            }

            // prepare for next word
            $index = 0; $word = "";
        }
    }

    // close file
    fclose($fp);

    // determine dictionary's size
    $before = microtime(TRUE);
    $n = size();
    $after = microtime(TRUE);

    // calculate time to determine dictionary's size
```

```
    $ti_size = $after - $before;

    // unload dictionary
    $before = microtime(TRUE);
    $unloaded = unload();
    $after = microtime(TRUE);

    // abort if dictionary not unloaded
    if (!$unloaded)
    {
        print("Could not load $dict.\n");
        return 5;
    }
    // calculate time to determine dictionary's size
    $ti_unload = $after - $before;

    // report benchmarks
    printf("\nWORDS MISSPELLED:     %d\n", $misspellings);
    printf("WORDS IN DICTIONARY:  %d\n", $n);
    printf("WORDS IN FILE:        %d\n", $words);
    printf("TIME IN load:         %.2f\n", $ti_load);
    printf("TIME IN check:        %.2f\n", $ti_check);
    printf("TIME IN size:         %.2f\n", $ti_size);
    printf("TIME IN unload:       %.2f\n", $ti_unload);
    printf("TOTAL TIME:           %.2f\n\n", $ti_load + $ti_check + $ti_size + $ti_unload);

?>
```