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

    /**
     * conditions-1.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Tells user if his or her input is positive, zero, or negative.
     *
     * Demonstrates use of if-else construct.
     */

    // ask user for an integer
    $n = readline("I'd like an integer please: ");

    // analyze user's input
    if ($n > 0)
    {
        printf("You picked a positive number!\n");
    }
    else if ($n == 0)
    {
        printf("You picked zero!\n");
    }
    else
    {
        printf("You picked a negative number!\n");
    }

?>
```

```php
<?php

    /**
     * counter.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a counter.  Demonstrates sessions.
     */

    // enable sessions
    session_start();

    // check counter
    if (isset($_SESSION["counter"]))
    {
        $counter = $_SESSION["counter"];
    }
    else
    {
        $counter = 0;
    }

    // increment counter
    $_SESSION["counter"] = $counter + 1;

?>

<!DOCTYPE html>

<html>
    <head>
        <title>counter</title>
    </head>
    <body>
        You have visited this site <?= $counter ?> time(s).
    </body>
</html>
```

```php
<?php

    /**
     * froshims-0.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-0.php.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body style="text-align: center;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-0.php" method="post">
            Name: <input name="name" type="text"/>
            <br/>
            <input name="captain" type="checkbox"/> Captain?
            <br/>
            <input name="gender" type="radio" value="F"/> Female
            <input name="gender" type="radio" value="M"/> Male
            <br/>
            Dorm:
            <select name="dorm">
                <option value=""></option>
                <option value="Apley Court">Apley Court</option>
                <option value="Canaday">Canaday</option>
                <option value="Grays">Grays</option>
                <option value="Greenough">Greenough</option>
                <option value="Hollis">Hollis</option>
                <option value="Holworthy">Holworthy</option>
                <option value="Hurlbut">Hurlbut</option>
                <option value="Lionel">Lionel</option>
                <option value="Matthews">Matthews</option>
                <option value="Mower">Mower</option>
                <option value="Pennypacker">Pennypacker</option>
                <option value="Stoughton">Stoughton</option>
                <option value="Straus">Straus</option>
                <option value="Thayer">Thayer</option>
                <option value="Weld">Weld</option>
```

```
                <option value="Wigglesworth">Wigglesworth</option>
            </select>
            <br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

```php
<?php

    /**
     * froshims-1.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-1.php.
     *
     * Demonstrates Bootstrap (http://getbootstrap.com/).
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"/>
        <title>Frosh IMs</title>
    </head>
    <body style="margin: 20px;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-1.php" method="post">
            <fieldset>
                <label>Name</label>
                <input name="name" type="text"/>
                <label class="checkbox">
                    <input name="captain" type="checkbox"/> Captain?
                </label>
                <label class="radio">
                    <input name="gender" type="radio" value="F"/> Female
                </label>
                <label class="radio">
                    <input name="gender" type="radio" value="M"/> Male
                </label>
                <label>
                    <select name="dorm">
                        <option value=""></option>
                        <option value="Apley Court">Apley Court</option>
                        <option value="Canaday">Canaday</option>
                        <option value="Grays">Grays</option>
                        <option value="Greenough">Greenough</option>
                        <option value="Hollis">Hollis</option>
                        <option value="Holworthy">Holworthy</option>
                        <option value="Hurlbut">Hurlbut</option>
```

```
                        <option value="Lionel">Lionel</option>
                        <option value="Matthews">Matthews</option>
                        <option value="Mower">Mower</option>
                        <option value="Pennypacker">Pennypacker</option>
                        <option value="Stoughton">Stoughton</option>
                        <option value="Straus">Straus</option>
                        <option value="Thayer">Thayer</option>
                        <option value="Weld">Weld</option>
                        <option value="Wigglesworth">Wigglesworth</option>
                    </select>
                </label>
                <button class="btn btn-default" type="submit">Register</button>
            </fieldset>
        </form>
    </body>
</html>
```

```php
<?php

    /**
     * froshims-2.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-2.php.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body style="text-align: center;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-2.php" method="post">
            Name: <input name="name" type="text"/>
            <br/>
            <input name="captain" type="checkbox"/> Captain?
            <br/>
            <input name="gender" type="radio" value="F"/> Female
            <input name="gender" type="radio" value="M"/> Male
            <br/>
            Dorm:
            <select name="dorm">
                <option value=""></option>
                <option value="Apley Court">Apley Court</option>
                <option value="Canaday">Canaday</option>
                <option value="Grays">Grays</option>
                <option value="Greenough">Greenough</option>
                <option value="Hollis">Hollis</option>
                <option value="Holworthy">Holworthy</option>
                <option value="Hurlbut">Hurlbut</option>
                <option value="Lionel">Lionel</option>
                <option value="Matthews">Matthews</option>
                <option value="Mower">Mower</option>
                <option value="Pennypacker">Pennypacker</option>
                <option value="Stoughton">Stoughton</option>
                <option value="Straus">Straus</option>
                <option value="Thayer">Thayer</option>
                <option value="Weld">Weld</option>
```

```
                <option value="Wigglesworth">Wigglesworth</option>
            </select>
            <br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

```php
<?php

    /**
     * froshims-3.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.
     * Submits to register-3.php.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body style="text-align: center;">
        <h1>Register for Frosh IMs</h1>
        <form action="register-3.php" method="post">
            Name: <input name="name" type="text"/>
            <br/>
            <input name="captain" type="checkbox"/> Captain?
            <br/>
            <input name="gender" type="radio" value="F"/> Female
            <input name="gender" type="radio" value="M"/> Male
            <br/>
            Dorm:
            <select name="dorm">
                <option value=""></option>
                <option value="Apley Court">Apley Court</option>
                <option value="Canaday">Canaday</option>
                <option value="Grays">Grays</option>
                <option value="Greenough">Greenough</option>
                <option value="Hollis">Hollis</option>
                <option value="Holworthy">Holworthy</option>
                <option value="Hurlbut">Hurlbut</option>
                <option value="Lionel">Lionel</option>
                <option value="Matthews">Matthews</option>
                <option value="Mower">Mower</option>
                <option value="Pennypacker">Pennypacker</option>
                <option value="Stoughton">Stoughton</option>
                <option value="Straus">Straus</option>
                <option value="Thayer">Thayer</option>
                <option value="Weld">Weld</option>
```

```
                <option value="Wigglesworth">Wigglesworth</option>
            </select>
            <br/>
            <input type="submit" value="Register"/>
        </form>
    </body>
</html>
```

```php
<?php

    /**
     * register-0.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Dumps contents of $_POST.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        <pre>
            <?php print_r($_POST); ?>
        </pre>
    </body>
</html>
```

```php
<?php

    /**
     * register-1.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a registration form for Frosh IMs.  Redirects
     * user to froshims-1.php upon error.
     */

    // validate submission
    if (empty($_POST["name"]) || empty($_POST["gender"]) || empty($_POST["dorm"]))
    {
        header("Location: http://localhost/src8m/froshims/froshims-1.php");
        exit;
    }

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        You are registered!  (Well, not really.)
    </body>
</html>
```

```php
<?php

    /**
     * register-2.php
     *
     * Computer Science 50
     * David J. Malan
     *
     * Implements a registration form for Frosh IMs.  Informs user of
     * any errors.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        <?php if (empty($_POST["name"]) || empty($_POST["gender"]) || empty($_POST["dorm"])): ?>
            You must provide your name, gender, and dorm!  Go <a href="froshims-2.php">back</a>.
        <?php else: ?>
            You are registered!  (Well, not really.)
        <?php endif ?>
    </body>
</html>
```

```php
<?php

    /**
     * register-3.php
     *
     * Computer Science 50
     * David J. Malan
     *
     * Implements a registration form for Frosh IMs.  Reports registration
     * via email.  Redirects user to froshims-3.php upon error.
     */

    // require PHPMailer
    require("libphp-phpmailer/class.phpmailer.php");

    // validate submission
    if (!empty($_POST["name"]) && !empty($_POST["gender"]) && !empty($_POST["dorm"]))
    {
        // instantiate mailer
        $mail = new PHPMailer();

        // use SMTP
        $mail->IsSMTP();
        $mail->Host = "smtp.fas.harvard.edu";
        $mail->Port = 587;
        $mail->SMTPSecure = "tls";

        // set From:
        $mail->SetFrom("jharvard@cs50.harvard.edu");

        // set To:
        $mail->AddAddress("jharvard@cs50.harvard.edu");

        // set Subject:
        $mail->Subject = "registration";

        // set body
        $mail->Body =
            "This person just registered:\n\n" .
            "Name: " . $_POST["name"] . "\n" .
            "Captain: " . $_POST["captain"] . "\n" .
            "Gender: " . $_POST["gender"] . "\n" .
            "Dorm: " . $_POST["dorm"];

        // send mail
        if ($mail->Send() == false)
        {
            die($mail->ErrInfo);
```

```
        }
    }
    else
    {
        header("Location: http://localhost/src8m/froshims/froshims-3.php");
        exit;
    }
?>

<!DOCTYPE html>

<html>
    <head>
        <title>Frosh IMs</title>
    </head>
    <body>
        You are registered!  (Really.)
    </body>
</html>
```

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

    /**
     * hello
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Says hello to the world.
     *
     * Demonstrates use of a shebang with env.
     */

    printf("hello, world\n");

?>
```

```php
<?php

    /**
     * dictionary.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a dictionary in a (non-object-oriented) way that
     * mimics Problem Set 6's implementation in C.  However, an
     * object-oriented design would be better in PHP.
     */

    // size of dictionary
    $size = 0;

    // hash table
    $table = [];

    /**
     * Returns true if word is in dictionary else false.
     */
    function check($word)
    {
        global $table;
        if (isset($table[strtolower($word)]))
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    /**
     * Loads dictionary into memory.  Returns true if successful else false.
     */
    function load($dictionary)
    {
        global $table, $size;
        if (!file_exists($dictionary) && is_readable($dictionary))
        {
            return false;
        }
        foreach (file($dictionary) as $word)
        {
            $table[chop($word)] = true;
```

```
            $size++;
        }
        return true;
    }

    /**
     * Returns number of words in dictionary if loaded else 0 if not yet loaded.
     */
    function size()
    {
        global $size;
        return $size;
    }

    /**
     * Unloads dictionary from memory.  Returns true if successful else false.
     */
    function unload()
    {
        return true;
    }

?>
```

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

    /**************************************************************************
     * speller.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Implements a spell-checker.
     **************************************************************************/

    require("dictionary.php");

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

    // default dictionary
    define("WORDS", "/home/cs50/pset5/dictionaries/large");

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

    // benchmarks
    $time_load = 0.0; $time_check = 0.0; $time_size = 0.0; $time_unload = 0.0;

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

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

    // abort if dictionary not loaded
    if (!$loaded)
    {
        print("Could not load $dictionary.\n");
        return 1;
    }

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

```
49.
50.     // try to open file
51.     $file = ($argc == 3) ? $argv[2] : $argv[1];
52.     $fp = fopen($file, "r");
53.     if ($fp === false)
54.     {
55.         print("Could not open $file.\n");
56.         return 1;
57.     }
58.
59.     // prepare to report misspellings
60.     printf("\nMISSPELLED WORDS\n\n");
61.
62.     // prepare to spell-check
63.     $word  = "";
64.     $index = 0; $misspellings = 0; $words = 0;
65.
66.     // spell-check each word in file
67.     for ($c = fgetc($fp); $c !== false; $c = fgetc($fp))
68.     {
69.         // allow alphabetical characters and apostrophes (for possessives)
70.         if (preg_match("/[a-zA-Z]/", $c) || ($c == "'" && $index > 0))
71.         {
72.             // append character to word
73.             $word .= $c;
74.             $index++;
75.
76.             // ignore alphabetical strings too long to be words
77.             if ($index >= LENGTH)
78.             {
79.                 // consume remainder of alphabetical string
80.                 while (($c = fgetc($fp)) !== false && preg_match("/[a-zA-Z]/", $c));
81.
82.                 // prepare for new word
83.                 $index = 0; $word = "";
84.             }
85.         }
86.
87.         // ignore words with numbers (like MS Word)
88.         else if (ctype_digit($c))
89.         {
90.             // consume remainder of alphabetical string
91.             while (($c = fgetc($fp)) !== false && preg_match("/[a-zA-z0-9]/", $c));
92.
93.             // prepare for new word
94.             $index = 0; $word = "";
95.         }
96.
```

```
        // 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
            $time_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
    $time_size = $after - $before;

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

    // abort if dictionary not unloaded
    if (!$unloaded)
    {
        print("Could not load $dictionary.\n");
        return 1;
    }
```

```
145.     // calculate time to determine dictionary's size
146.     $time_unload = $after - $before;
147.
148.     // report benchmarks
149.     printf("\nWORDS MISSPELLED:     %d\n", $misspellings);
150.     printf("WORDS IN DICTIONARY:  %d\n", $n);
151.     printf("WORDS IN TEXT:        %d\n", $words);
152.     printf("TIME IN load:         %.2f\n", $time_load);
153.     printf("TIME IN check:        %.2f\n", $time_check);
154.     printf("TIME IN size:         %.2f\n", $time_size);
155.     printf("TIME IN unload:       %.2f\n", $time_unload);
156.     printf("TOTAL TIME:           %.2f\n\n", $time_load + $time_check + $time_size + $time_unload);
157.
158. ?>
```

```php
<?php

    /**
     * index.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * A home page for the course.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>CS50</title>
    </head>
    <body>
        <h1>CS50</h1>
        <ul>
            <li><a href="lectures.php">Lectures</a></li>
            <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/syllabus/syllabus.html">Syllabus</a></li>
        </ul>
    </body>
</html>
```

```php
<?php

    /**
     * lectures.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Links to lectures.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Lectures</title>
    </head>
    <body>
        <h1>Lectures</h1>
        <ul>
            <li><a href="week0.php">Week 0</a></li>
            <li><a href="week1.php">Week 1</a></li>
        </ul>
    </body>
</html>
```

```
1. mvc/0/README
2.
3. David J. Malan
4. malan@harvard.edu
5.
6. index.php - home page for course
7. lectures.php - a list of weeks
8. week0.php - a week
9. week1.php - a week
```

```php
<?php

    /**
     * week0.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Represents Week 0.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Week 0</title>
    </head>
    <body>
        <h1>Week 0</h1>
        <ul>
            <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/week0w.pdf">Wednesday</a></li>
            <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/f/week0f.pdf">Friday</a></li>
        </ul>
    </body>
</html>
```

```php
<?php

    /**
     * week1.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Represents Week 1.
     */

?>

<!DOCTYPE html>

<html>
    <head>
        <title>Week 1</title>
    </head>
    <body>
        <h1>Week 1</h1>
        <ul>
            <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/m/week1m.pdf">Monday</a></li>
            <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/w/week1w.pdf">Wednesday</a></li>
        </ul>
    </body>
</html>
```

```
    </body>
</html>
```

```
<!DOCTYPE html>

<html>
    <head>
        <title>CS50</title>
    </head>
    <body>
        <h1>CS50</h1>
```

```php
<?php require("header.php"); ?>

<ul>
    <li><a href="lectures.php">Lectures</a></li>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/syllabus/syllabus.html">Syllabus</a></li>
</ul>

<?php require("footer.php"); ?>
```

```php
<?php require("header.php"); ?>

<ul>
    <li><a href="week0.php">Week 0</a></li>
    <li><a href="week1.php">Week 1</a></li>
</ul>

<?php require("footer.php"); ?>
```

```
mvc/1/README

David J. Malan
malan@harvard.edu

Improves upon mvc/0 by factoring out pages' header and footer.

footer.php - pages' footer
header.php - pages' header
index.php - home page for course
lectures.php - a list of weeks
week0.php - a week
week1.php - a week
```

```
1. <?php require("header.php"); ?>
2.
3. <ul>
4.   <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/week0w.pdf">Wednesday</a></li>
5.   <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/f/week0f.pdf">Friday</a></li>
6. </ul>
7.
8. <?php require("footer.php"); ?>
```

```
1. <?php require("header.php"); ?>
2.
3. <ul>
4.     <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/m/week1m.pdf">Monday</a></li>
5.     <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/w/week1w.pdf">Wednesday</a></li>
6. </ul>
7.
8. <?php require("footer.php"); ?>
```

```
    </body>
</html>
```

```php
<!DOCTYPE html>

<html>
    <head>
        <title><?= htmlspecialchars($title) ?></title>
    </head>
    <body>
        <h1><?= htmlspecialchars($title) ?></h1>
```

```php
<?php

    /**
     * Renders footer.
     */
    function renderFooter($data = [])
    {
        extract($data);
        require("footer.php");
    }

    /**
     * Renders header.
     */
    function renderHeader($data = [])
    {
        extract($data);
        require("header.php");
    }

?>
```

```php
<?php require("helpers.php"); ?>

<?php renderHeader(["title" => "CS50"]); ?>

<ul>
    <li><a href="lectures.php">Lectures</a></li>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/syllabus/syllabus.html">Syllabus</a></li>
</ul>

<?php renderFooter(); ?>
```

```php
<?php require("helpers.php"); ?>

<?php renderHeader(["title" => "Lectures"]); ?>

<ul>
    <li><a href="week0.php">Week 0</a></li>
    <li><a href="week1.php">Week 1</a></li>
</ul>

<?php renderFooter(); ?>
```

```
mvc/2/README

David J. Malan
malan@harvard.edu

Improves upon mvc/1 by wrapping header and footer with a parameterized function.

footer.php - pages' footer
header.php - pages' header
helpers.php - helper functions
index.php - home page for course
lectures.php - a list of weeks
week0.php - a week
week1.php - a week
```

```php
<?php require("helpers.php"); ?>

<?php renderHeader(["title" => "Week 0"]); ?>

<ul>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/week0w.pdf">Wednesday</a></li>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/f/week0f.pdf">Friday</a></li>
</ul>

<?php renderFooter(); ?>
```

```php
<?php require("helpers.php"); ?>

<?php renderHeader("title" => "Week 1"]); ?>

<ul>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/m/week1m.pdf">Monday</a></li>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/w/week1w.pdf">Wednesday</a></li>
</ul>

<?php renderFooter(); ?>
```

```
1.      </body>
2.  </html>
```

```php
<!DOCTYPE html>

<html>
    <head>
        <title><?= htmlspecialchars($title) ?></title>
    </head>
    <body>
        <h1><?= htmlspecialchars($title) ?></h1>
```

```php
<?php

    /**
     * Renders template.
     */
    function render($template, $data = [])
    {
        $path = $template . ".php";
        if (file_exists($path))
        {
            extract($data);
            require($path);
        }
    }

?>
```

```
1. <?php require("helpers.php"); ?>
2.
3. <?php render("header", ["title" => "CS50"]); ?>
4.
5. <ul>
6.     <li><a href="lectures.php">Lectures</a></li>
7.     <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/syllabus/syllabus.html">Syllabus</a></li>
8. </ul>
9.
10. <?php render("footer"); ?>
```

```php
<?php require("helpers.php"); ?>

<?php render("header", ["title" => "Lectures"]); ?>

<ul>
    <li><a href="week0.php">Week 0</a></li>
    <li><a href="week1.php">Week 1</a></li>
</ul>

<?php render("footer"); ?>
```

```
mvc/3/README

David J. Malan
malan@harvard.edu

Improves upon mvc/2 by generalizing header and footer as templates.

footer.php - pages' footer
header.php - pages' header
helpers.php - helper functions
index.php - home page for course
lectures.php - a list of weeks
week0.php - a week
week1.php - a week
```

```
1.  <?php require("helpers.php"); ?>
2.
3.  <?php render("header", ["title" => "Week 0"]); ?>
4.
5.  <ul>
6.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/week0w.pdf">Wednesday</a></li>
7.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/f/week0f.pdf">Friday</a></li>
8.  </ul>
9.
10. <?php render("footer"); ?>
```

```
<?php require("helpers.php"); ?>

<?php render("header", ["title" => "Week 1"]); ?>

<ul>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/week1m.pdf">Monday</a></li>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/week1w.pdf">Wednesday</a></li>
</ul>

<?php render("footer"); ?>
```

```php
<?php

    /**
     * Renders template.
     */
    function render($template, $data = [])
    {
        $path = __DIR__ . "/../templates/" . $template . ".php";
        if (file_exists($path))
        {
            extract($data);
            require($path);
        }
    }

?>
```

```
1.  <?php require("includes/helpers.php"); ?>
2.
3.  <?php render("header", ["title" => "CS50"]); ?>
4.
5.  <ul>
6.      <li><a href="lectures.php">Lectures</a></li>
7.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/syllabus/syllabus.html">Syllabus</a></li>
8.  </ul>
9.
10. <?php render("footer"); ?>
```

```php
<?php require("includes/helpers.php"); ?>

<?php render("header", ["title" => "Lectures"]); ?>

<ul>
    <li><a href="week0.php">Week 0</a></li>
    <li><a href="week1.php">Week 1</a></li>
</ul>

<?php render("footer"); ?>
```

```
mvc/4/README

David J. Malan
malan@harvard.edu

Improves upon mvc/3 by organizing files into subdirectories.

includes/
   helpers.php - helper functions
index.php - home page for course
lectures.php - a list of lectures
templates/
   footer.php - pages' footer
   header.php - pages' header
week0.php - a week
week1.php - a week
```

```
    </body>
</html>
```

```
1. <!DOCTYPE html>
2.
3. <html>
4.     <head>
5.         <title><?= htmlspecialchars($title) ?></title>
6.     </head>
7.     <body>
8.         <h1><?= htmlspecialchars($title) ?></h1>
```

```
1.  <?php require("includes/helpers.php"); ?>
2.
3.  <?php render("header", ["title" => "Week 0"]); ?>
4.
5.  <ul>
6.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/week0w.pdf">Wendesday</a></li>
7.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/f/week0f.pdf">Friday</a></li>
8.  </ul>
9.
10. <?php render("footer"); ?>
```

```
1. <?php require("includes/helpers.php"); ?>
2.
3. <?php render("header", ["title" => "Week 1"]); ?>
4.
5. <ul>
6.     <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/week1m.pdf">Monday</a></li>
7.     <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/week1w.pdf">Wednedsay</a></li>
8. </ul>
9.
10. <?php render("footer"); ?>
```

```php
<?php

    /**
     * Renders template.
     */
    function render($template, $data = [])
    {
        $path = __DIR__ . "/../templates/" . $template . ".php";
        if (file_exists($path))
        {
            extract($data);
            require($path);
        }
    }

?>
```

```php
<?php require("../includes/helpers.php"); ?>

<?php render("header", ["title" => "CS50"]); ?>

<ul>
    <li><a href="lectures.php">Lectures</a></li>
    <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/syllabus/syllabus.html">Syllabus</a></li>
</ul>

<?php render("footer"); ?>
```

```php
<?php require("../includes/helpers.php"); ?>

<?php render("header", ["title" => "Lectures"]); ?>

<ul>
    <li><a href="week0.php">Week 0</a></li>
    <li><a href="week1.php">Week 1</a></li>
</ul>

<?php render("footer"); ?>
```

```
1.  <?php require("../includes/helpers.php"); ?>
2.
3.  <?php render("header", ["title" => "Week 0"]); ?>
4.
5.  <ul>
6.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/w/week0w.pdf">Wednesday</a></li>
7.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/0/f/week0f.pdf">Friday</a></li>
8.  </ul>
9.
10. <?php render("footer"); ?>
```

```
1.  <?php require("../includes/helpers.php"); ?>
2.
3.  <?php render("header", ["title" => "Week 1"]); ?>
4.
5.  <ul>
6.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/m/week1m.pdf">Monday</a></li>
7.      <li><a href="http://cdn.cs50.net/2014/fall/lectures/1/w/week1w.pdf">Wednesday</a></li>
8.  </ul>
9.
10. <?php render("footer"); ?>
```

```
mvc/5/README

David J. Malan
malan@harvard.edu

Improves upon mvc/4 by isolating web-accessible content in its own subdirectory.

html/
    index.php - home page for course
    lectures.php - a list of lectures
    week0.php - a week
    week1.php - a week
includes/
    helpers.php - helper functions
templates/
    footer.php - pages' footer
    header.php - pages' header
```

```
    </body>
</html>
```

```
1. <!DOCTYPE html>
2.
3. <html>
4.     <head>
5.         <title><?= htmlspecialchars($title) ?></title>
6.     </head>
7.     <body>
8.         <h1><?= htmlspecialchars($title) ?></h1>
```

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

    /**
     * return.php
     *
     * David J. Malan
     * malan@harvard.edu
     *
     * Cubes a variable.
     *
     * Demonstrates use of parameter and return value.
     */

    $x = 2;
    printf("x is now %d\n", $x);
    printf("Cubing...\n");
    $x = cube($x);
    printf("Cubed!\n");
    printf("x is now %d\n", $x);

    /**
     * Cubes argument.
     */
    function cube($n)
    {
        return $n * $n * $n;
    }

?>
```