

COMP519 Practical 14

PHP (4)

Introduction

- This worksheet contains further exercises that are intended to familiarise you with PHP Programming. In particular, we want consider the problem of maintaining a ‘program state’ during a sequence of interactions with a user.

While you work through the exercises below compare your results with those of your fellow students and ask for help and comments if required.

- You might proceed more quickly if you cut-and-paste code from this PDF file. Note that a cut-and-paste operation may introduce extra spaces into your code. It is important that those are removed and that your code exactly matches that shown in this worksheet.
- The exercises and instructions in this worksheet assume that you use the Department’s Linux systems to experiment with PHP.
- To keep things simple, we will just use a text editor, a terminal, and a web browser. You can use whatever text editor and web browser you are most familiar or comfortable with.

Exercises

1. In the first exercise we replicate an example from the lecture notes.

a. Create a file `php14A.php` in your `public_html` directory with the following content:

```
<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP14 A</title>
  </head>
  <body>
<?php
  echo '
    <form action="php14B.php" method="post">
      <label>Item: <input type="text" name="item"></label>
    </form>
  ';
?>
  </body>
</html>
```

b. Create a file `php14B.php` in your `public_html` directory with the following content:

```
<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP14 B</title>
  </head>
  <body>
```

```

<?php
echo 'Item: ', $_REQUEST['item'], "<br>";
echo '
  <form action="php14C.php" method="post">
    <label>Address: <input type="text" name="address"></label>
  </form>
  ';
?>
</body>
</html>

```

- c. Create a file php14C.php in your public_html directory with the following content:

```

<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP14 C</title>
  </head>
  <body>
    <?php
echo 'Item: ', $_REQUEST['item'], '<br>';
echo 'Address: ', $_REQUEST['address'], '<br>';
?>
  </body>
</html>

```

- d. Make sure that the file permissions of all three PHP scripts are set correctly and that they are free of syntax errors, then open

<https://student.csc.liv.ac.uk/~<user>/php14A.php>

where you replace *<user>* by your own username.

You should see a web page with an input field (Figure 1a). Enter something into that input field, say, Apple, and press RETURN. (In the following we call this ‘Item’ information.) You should then be taken to php14B.php (Figure 1b). You should see what you have entered on php14A.php, and there is another input field. Enter something into that input field, say, Liverpooli and press RETURN. (In the following we call this ‘Address’ information.) This takes you to php14B.php (Figure 1c). Ideally, you would like to see both what you have entered on php14A.php and what you have entered php14B.php, but only the latter is shown.

- e. Modify the file php14B.php, and php14C.php if necessary, so that the information from both input fields is transferred to php14C.php and the output produced by php14C.php is as shown in Figure 1d.

You might first try *a hidden input* and then *a session variable* to do so.

Item:

(a) php14A.php

Item: Apple

Address:

(b) php14B.php

Item:
Address: Liverpool

(c) php14C.php (Initial)

Item: Apple
Address: Liverpool

(d) php14C.php (Final)

Figure 1: Sequence of Web Pages

2. That the code and process of entering two pieces of information is spread across three files, corresponding to the three stages of the interaction between the script and the user, is rather awkward and makes maintaining the code difficult.

a. Create a file php14D.php in your public_html directory with the following content:

```
<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP14 D</title>
  </head>
  <body>
<?php
// First stage: Entry of `Item' information
function formItem() {
    echo '<form action="php14D.php" method="post">
        <label>Item: <input type="text" name="item"></label>
    </form>';
}

// Second stage: Entry of `Address' information
function formAddress() {
    echo '<form action="php14D.php" method="post">';
    echo ' <label>Address: <input type="text" name="address"></label>
    </form>';
}

// Third stage: Processing of `Item' and `Address' information
function processInputs() {
    echo 'Item: ', $_REQUEST['item'], '<br>';
    echo 'Address: ', $_REQUEST['address'], '<br>';
}

if (FALSE) {
    // Executing third stage
    processInputs();
} elseif (FALSE) {
    // Executing second stage
    formAddress();
}
```

```

} else {
    // Executing first stage
    formItem();
}
?>
</body>
</html>

```

- b. Make sure that the file permissions of `php14D.php` are set correctly and that they are free of syntax errors, then open

`https://student.csc.liv.ac.uk/~<user>/php14D.php`

where you replace `<user>` by your own username. Right now you should see a web page with an input field (Figure 1a) just as the one produced by `php14A.php`.

- c. Extend the code in `php14D.php` in the same way you did in Exercise 1e.
- d. Now modify the conditions in the conditional statements so that
- if the user has entered 'Item' information, but not 'Address' information, then the function `formAddress()` should be executed;
 - if the user has entered both 'Item' information and 'Address' presses RETURN, then the function `processInputs()` should be executed.

Test your solution.

- e. Does it make a difference whether the user enters a non-empty string or just presses RETURN in the input field?
- f. Modify the conditional statement so that only entry of a non-empty string moves the process to the next stage.
- g. Modify the definition of the `formItem()` function so that if `$_REQUEST['item']` is set and is an empty string, then an error message is shown below the input field telling the user to enter a non-empty string.
- h. Modify the definition of the `formAddress()` function so that if `$_REQUEST['address']` is set and is an empty string, then an error message is shown below the input field telling the user to enter a non-empty string.
3. Let us try the same three stage progress but with interdependent drop-down menus. Here we want the user to first select a sport from a drop-down menu and then a team for that sport from a second drop-down menu. In the final stage we again just print out the information the user has selected.

- a. Create a file `php14E.php` in your `public_html` directory with the following content:

```

<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP14 E</title>
  </head>
  <body>
<?php

// First stage: Select a sport
function selectSport() {

```

```

$sports = array('Football','Rugby');
echo '<form action="php14E.php" method="post">
    <label>Sport:</label>
    <select name="sport">
        <option value="">Select a sport</option>';
// foreach loop here
echo ' </select>
    <input type="submit">
</form>';
}

// Second stage: Select a team
function selectTeam() {
    $teams = array('Football' => array('Arsenal','Liverpool'),
        'Rugby' => array('St Helens','Warrington'));
    echo 'Sport: ', $_REQUEST['sport'], '<br>';
    echo '<form action="php14E.php" method="post">';
    echo ' <label>Team:</label>
        <select name="team">
            <option value="">Select a team</option>';
    // foreach loop here
    echo ' </select>
        <input type="submit">
        </form>';
}

// Third stage: Processing of `Sport' and `Team' information
function processInputs() {
    echo 'Sport: ', $_REQUEST['sport'], '<br>';
    echo 'Team: ', $_REQUEST['team'], '<br>';
}

if (FALSE) {
    // Executing third stage
    processInputs();
} elseif (FALSE) {
    // Executing second stage
    selectTeam();
} else {
    // Executing first stage
    selectSport();
}
?>
</body>
</html>

```

- b. Make sure that the file permissions of php14E.php are set correctly and that they are free of syntax errors, then open

<https://student.csc.liv.ac.uk/~<user>/php14E.php>

where you replace <user> by your own username. You should see the drop-down menu

for the first stage, though that drop-down menu only has one entry.

- c. Extend the code in `php14E.php` in the same way you did in Exercise 1e.
- d. Now modify the conditions in the conditional statements so that
 - if the user has selected a sport information, but has not selected a team, then the function `selectTeam()` should be executed;
 - if the user has selected a sport and a team, then the function `processInputs()` should be executed.
- e. Modify the definition of the `selectSport()` function so that the first drop-down menu is filled with options corresponding to the elements of the array `$sports`.
- f. Modify the definition of the `selectTeam()` function so that the second drop-down menu is filled with options corresponding to the elements of the array `$teams` for the sport that the user has selected in the first drop-down menu.
- g. Test whether your code works correctly. If it does, then in the third stage of the process the script will show you the sport and the team that the user has selected.