

7.3 Activity 1: Module 7 practical exercises 1-2

Purpose

In this module's exercises, you will write PHP scripts that correctly retrieve data from inputs via post and get methods, and write PHP scripts that correctly retrieve data from inputs using the basic ideas of postback. You will need to implement a simple map using the Google Maps API and utilise features such as markers, labels and Info Windows in the map.



From this module, Exercises 2 and 3 will be assessed as part of [Assignment 2B: Practical exercise set 2 \(https://canvas.westernsydneyonline.edu.au/courses/1261/assignments/15052\)](https://canvas.westernsydneyonline.edu.au/courses/1261/assignments/15052).

Task

Exercise 1

Step 1: In the practicals folder of your TWA website.

Step 2: Download the following zip file and then upload the HTML and CS files into your TWA website in the practicals/week8 folder.

- [Module 7 Windows files \(ZIP 3 KB\)](https://canvas.westernsydneyonline.edu.au/courses/1261/files/882000?wrap=1)
(<https://canvas.westernsydneyonline.edu.au/courses/1261/files/882000?wrap=1>)_ 
(https://canvas.westernsydneyonline.edu.au/courses/1261/files/882000/download?download_frd=1) .
- [Module 7 MACOSX files \(ZIP 3 KB\)](https://canvas.westernsydneyonline.edu.au/courses/1261/files/882001?wrap=1)
(<https://canvas.westernsydneyonline.edu.au/courses/1261/files/882001?wrap=1>)_ 
(https://canvas.westernsydneyonline.edu.au/courses/1261/files/882001/download?download_frd=1) .

Step 3: Create a file named exercises1.php in the practicals/week8 folder on your TWA website.

Step 4: Apply the following code to exercise1.php.

```
<!DOCTYPE html>
<html lang="en"
<head>
  <meta charset="utf-8"
  <title>Week 7 Exercise 1</title>
</head>
```

```

<body>
<?php
    //obtain the first name input from the $_GET array
    $namestr = $GET["firstname"];
    //obtain the values for the other input devices here

?>
<p>The following information was received from the form:</p>
<p><strong>name = </strong> <?php echo "$namestr"; ?></p>
<!--output the other form inputs here -->
</body>
</html>

```

Step 5: Test the PHP script by submitting data from exercise1.html to the PHP script. You will notice that not all of the input devices from the form are shown in the output of the PHP page. Fix this by making appropriate modifications to the PHP file so that all input devices are shown as part of the output.

Note: What you should notice by completing this exercise is that it does not matter what type of input device is used on the form (text box, radio button, checkbox, selection list, etc) the method of obtaining the values is the same.

Exercise 2

Step 1: Upload the exercise2.html from the zip file to your practicals/week8 holder onto your website.

The forms in Exercise 1 and Exercise 2 are very similar but with some important differences. These include:

- the selection list size has been changed to '4'
- multiple selections are now enabled in the selection list
- the action of the form has been changed to exercise2.php
- the form method has been changed.

Step 2: Create a file named exercise2.php in the practicals/week8 holder onto your website.

Step 3: Copy your solution code from exercise1.php into exercise2.php.

Step 4: Modify exercise2.php to correctly process the form from exercise2.html.

Hint: Because of the changes made to the form (in particular, the change to the form method is critical) you will need to modify several aspects of the PHP script.

Step 5: Ensure that your exercise2 script is capable of displaying all of the selections made from the selection list—it won't without appropriate modifications to both the HTML form and the PHP.

Hint: Review lecture slides and examples for how to process input devices that can have multiple values.