

COMP519 Practical 4

PHP (3)

Introduction

- This worksheet contains exercises that are intended to familiarise you with PHP Programming. 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 the 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.
- If you do not manage to get through all the exercises during this practical session, please complete them in your own time before the next practical takes place.

Exercises

1. The following exercise deals with *pattern matching*.
 - a. Use a text editor to create a file php04A.php in your public_html directory with the following content:

```
<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP 04A</title>
  </head>
  <body>
    <h1>Pattern Matching</h1>
  <?php
echo "<h2>Exercise 1</h2>\n";
$words = ["p0wned","l33t", "hello_world", "reg-123", // words
          "WWW", "ClickMe", "'twas", "o'clock", "a", // words
          "has'", "O'Donnell", "'s", "s'", // words
          "10", "123-reg", "even-", "a@b", "a:b", // not words
          "evenin'", "hal''lo", "be'-tter" // not words
];
foreach ($words as $word) {
  echo "Word: $word ";
  if (isWord($word)) {
    echo "is a word";
  } else {
```

```

    echo "is not a word";
  }
  echo "<br>\n";
}

function isWord($word) {
}
?>
</body>
</html>

```

Make sure that the file permission of the file are correct.

- b. Open a web browser and access the URL

`https://student.csc.liv.ac.uk/~<user>/php04A.php`

were `<user>` should be replaced by your user name. At the moment, the script only prints out all the strings in `$words` and claims all of them are not words.

- c. As the comments in the code indicate, not all the strings in `$words` are words. Only sequences of characters that (i) only consist of ASCII letters, ASCII digits, apostrophes, hyphens and underscores; (ii) start with a letter or start with an apostrophe followed by a letter, (iii) do not contain a sequence of two or more apostrophes or hyphens, and (iv) end with (a) a letter, (b) a digit, or (c) an apostrophe preceded by the letter s, are words.

Complete the definition of the function `isWord()` so that it returns true for arguments that are words and false otherwise. Use the PHP function `preg_match` with appropriate regular expressions for this. It's OK to use more than one call of `preg_match` but four should really be sufficient.

If correct, your definition of `isWord()` should classify words as indicated by the comments in the code.

2. The following exercise deals with the *modification of array elements, string operations and pattern search and replace*.

- a. Add the following PHP code to the end of the current PHP code in `php04A.php`:

```

echo "<h2>Exercise 2</h2>\n";
$names = ["Dave Shield", "Mr Andy Roxburgh",
          "Mr Micheal Abaho",
          "Dr Ullrich Hustadt", "Prof Boris Konev"];
foreach ($names as $name)
  echo "(1) Name: $name<br>\n";
// Your own code here
foreach ($names as $name)
  echo "(2) Name: $name<br>\n";

```

Save the file and reload the page in your web browser. The output produced by the code above should be as follows:

```

(1) Name: Dave Shield
(1) Name: Mr Andy Roxburgh
(1) Name: Mr Micheal Abaho
(1) Name: Dr Ullrich Hustadt
(1) Name: Prof Boris Konev

```

and the same again with (1) replaced by (2).

- b. Add code at the point indicated in the code in Exercise 2a that modifies the strings stored in \$names so that the output produced will change to:

```
(2) Name: SHIELD, Dave  
(2) Name: ROXBURGH, Andy  
(2) Name: ABAHO, Micheal  
(2) Name: HUSTADT, Ullrich  
(2) Name: KONEV, Boris
```

This should be done with the help of string manipulation operations including `preg_match` or `preg_replace_callback`, and the code should be independent of the actual names stored in \$names.

Hint: Recall from the lecture notes the variation of a `foreach`-loop that allows you to modify the elements of an array.