

COMP519 Practical 12

PHP (2)

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 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.
- 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. In the first exercise we want to explore how *variables* and *constants* behave in PHP.
 - a. Open a text editor and enter the following HTML markup and PHP code below, and save it as a file php12A.php in \$HOME/public_html/.

```
<!DOCTYPE html>
<html lang='en-GB'>
  <head>
    <title>PHP 12A</title>
  </head>
  <body>
    <h1>Variables and Constants</h1>
  <?php
    error_reporting( E_ALL );
    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    echo "<h2>Exercise 1a</h2>\n";
    echo "(1) The value of \$userAge is: $userAge<br>\n";
    echo "(2) The value of \$userAge is: ",var_dump($userAge),"<br>\n";
    echo "(3) This statement is executed<br>\n";
    $userAge = 27;
    echo "(4) The value of \$userAge is: $userAge<br>\n";
  ?>
</body>
</html>
```

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

- b. Open a web browser and access the URL

`https://student.csc.liv.ac.uk/~<user>/php12A.php`

were `<user>` should be replaced by your user name. You should see the output produced by the PHP script, including PHP notices indicating that in the first and second echo statement the variable `$userAge` is undefined. The output of `var_dump` still shows you what the value `$userAge` is at that point. Compare that with the behaviour of `jsDemo06A.html` that you have created for Practical 6.

- c. Add the following code at the end of the script:

```
echo "<h2>Exercise 1c</h2>\n";
if (FALSE) {
    $x = 519;
} else {
    echo "(1) The value of \$x is $x<br>\n";
    echo "(2) The value of \$x is ",var_dump($x),"<br>\n";
}
```

Save the file and reload the page in your web browser. You should see PHP notices that the variable `$x` is undefined.

2. The following example deals with the use of *constants* in PHP code.

- a. Add the following PHP code to the end of the current PHP code in `php12A.php` (make sure that the additional PHP code is included before `??`):

```
echo "<h2>Exercise 2</h2>\n";
define("PI",3.14159);
define("SPEED_OF_LIGHT",299792458,true);
print "1 - Value of PI: PI<br>\n";
print "2 - Value of PI: ".PI."<br>\n";
$diameter = 2;
$time     = 3;
$circumference1 = PI * $diameter;
# The next line works neither in PHP5 nor PHP7
$circumference2 = pi * $diameter;
# The next line produces a warning in PHP7 but not PHP5
$distance = speed_of_light * $time;
echo "Diameter = $diameter => ",
    "Circumference1 = $circumference1 | ",
    "Circumference2 = $circumference2<br>\n";
echo "Time     = $time => Distance = $distance<br>\n";
```

- b. Save the file and execute the script again using the web browser.
c. Check that the additional output shown in the web browser starts with the two lines

```
1 - Value of PI: PI
2 - Value of PI: 3.14159
```

and that you understand how the difference between lines 1 and 2 comes about.

- d. Make sure that you understand how the constants `PI` and `SPEED_OF_LIGHT` 'work' and why the output indicates that `Circumference1` has value 6.28318 while `Circumference2` has value 0.

- e. Change your script so that the four lines of output produced by the code in Exercise 2a appear in a red bold-face font in the web browser.

3. This exercise is about *arrays* and *array operators* in PHP.

- a. In the lectures we have seen that PHP has array operators that allow us to use arrays as stacks or queues. Let us replicate an example that we have seen in JavaScript using PHP.

Open a text editor, enter the following HTML markup and PHP code, and save it to a file name php12B.php in \$HOME/public_html/. Make sure that the file permission of the file are correct.

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP 12B</title>
  </head>
  <body>
    <h1>Array Operators and Regular Expressions</h1>
    <?php
echo "<h2>Exercise 3a</h2>\n";
$planets = array("earth");
array_unshift($planets,"mercury","venus");
array_push($planets,"mars","jupiter","saturn");
echo "(1) \$planets = [" ,join(" ,",$planets),"]<br>\n";
$last = array_pop($planets);
echo "(2) \$planets = [" ,join(" ,",$planets),"]<br>\n";
$first = array_shift($planets);
echo "(3) \$planets = [" ,join(" ,",$planets),"]<br>\n";
echo "(4) \$first = $first, \$last = $last<br>\n";
?>
    </body>
</html>
```

- b. Open a web browser and access the URL

<https://student.csc.liv.ac.uk/~<user>/php12B.php>

were <user> should be replaced by your user name.

You should see the output produced by the PHP script showing you how the various array operators change an array. Make sure that you understand how each operator works.

- c. Assignments involving arrays work differently in PHP than in JavaScript and Python. To verify that add the following code at the end of the PHP script in php12B.php:

```
echo "<h2>Exercise 3c</h2>\n";
$spheres = $planets;
echo "(5) \$spheres = [" ,join(" ,",$spheres),"]<br>\n";
$planets[1] = "midgard";
echo "(6) \$planets = [" ,join(" ,",$planets),"]<br>\n";
echo "(6) \$spheres = [" ,join(" ,",$spheres),"]<br>\n";
$spheres = &$planets;
echo "(7) \$spheres = [" ,join(" ,",$spheres),"]<br>\n";
```

```
$planets[0] = "friga";
echo "(8) \$planets = [",join(", ", $planets),"]<br>\n";
echo "(8) \$spheres = [",join(", ", $spheres),"]<br>\n";
```

Save the file and reload the page in your web browser. What does the output tell us about how array assignments work in PHP?

- d. Extend the script with code that repeatedly removes elements from the front of the array `$planets` until the array is empty and in each step prints out the element that was removed as well as the elements that remain in the array. The output should look as follows:

```
Removed: friga Remaining: [midgard, mars, jupiter]
Removed: midgard Remaining: [mars, jupiter]
Removed: mars Remaining: [jupiter]
Removed: jupiter Remaining: []
```

4. In PHP there is no difference between an array and a hash / dictionary. All arrays are *associative arrays*.

- a. Create a new web page called `php12C.php` with the following content

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP 12C</title>
  </head>
  <body>
    <h1>Associative Arrays</h1>
  <?php
    $dict1 = array('a' => 1, 'b' => 2);
    $dict2 = $dict1;
    $dict1['b'] = 4;
    echo "\$dict1['b'] = ", $dict1['b'], "<br>\n";
    echo "\$dict2['b'] = ", $dict2['b'], "<br>\n";
  ?>
  </body>
</html>
```

Open the web page in your web browser. Compare the behaviour exhibited by PHP with that of JavaScript, say in `jsDemo08C.html` in Practical 8.

- b. Add code for a loop that prints out all the key-value pairs in `$dict1`.
- c. A possible application of a dictionary is to count the number of occurrences of particular strings. For example, `$dict1` may be seen to store the information that the character 'a' has one occurrence and the character 'b' four occurrences in some text. We want to develop code that performs such a count.

Add the following code at the end of your script

```
$text = 'lorem ipsum elit congue auctor inceptos taciti suscipit
tortor auctor integer congue hac nullam hac auctor tellus nullam
inceptos nullam integer tellus nullam auctor elit lorem ipsum elit';
```

Add code to your script that uses an associative array `$dict3` to count the number of occurrences of each word in this string and then prints a list of all the different words and their number of occurrences.

If your code is correct, the output will be something like

```
lorem -> 2
ipsum -> 2
elit -> 3
congue -> 2
auctor -> 4
inceptos -> 2
taciti -> 1
suscipit -> 1
tortor -> 1
integer -> 2
hac -> 2
nullam -> 4
tellus -> 2
```

- d. Modify the code that prints out the key-value pairs in `$dict3` so that the key-value pairs are shown in order of the values starting with the highest value. The output should take the form of a HTML table similar to the following:

Word	No of Occurrences
nullam	4
auctor	4
elit	3
tellus	2
ipsum	2
lorem	2
integer	2
hac	2
congue	2
inceptos	1
taciti	1
inceptos	1
suscipit	1
tortor	1

Hint: See the PHP Manual on array sorting at

<http://php.net/manual/en/array.sorting.php>.