Lab 3 – Strings

Task 1: Understanding string functions (5 marks)

Step 1:
Create a file **`strprocess.php`** that will receive an input from **`strform.html`** from Step 2 via POST method, remove all the vowels then output the resulting string. It should check if the input contains only the letters and spaces using regular expression, otherwise, it should generate an appropriate error message.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>Using string functions</title>
</head>
<body>
    <h1>Web Development – Lab 3</h1>
    <?php
       if (isset ($_POST["___(1)___"]) { // check if form data exists
         $str = $_POST["___(2)___"];     // obtain the form data
         $pattern = "/^[A-Za-z ]+$/";    // set regular expression pattern
         if (_______(3)_______) {        // check if $str with regular expression
           $ans = "";                    // initialise variable for the answer
           $len = _____(4)_____;         // obtain length of string $str
           for ($i = 0; $i < $len; $i++) {      // checks all characters in $str
             $letter = substr ($str, (5), 1);   // extract 1 char using substr
             // check using strops, is_numeric is used as strops returns a number
             // (position) if found, and false otherwise
             if (! is_numeric (strpos ("AEIOUaeiou", (6)))){
                $ans = $ans . $letter;          // concatenate letter to answer
             }
           }
           // generate answer after all letters are checked
           echo "<p>The word with no vowels is ", $ans, ".</p>";
         } else {                        // string contains invalid characters
           echo "<p>Please enter a string containing only letters or space.</p>";
         }
       } else {                          // no input
         echo "<p>Please enter string from the input form.</p>";
       }
    ?>
</body>
</html>
```

Step 2:
Create a file **`strform.html`** that contains a form with a single text box that allows a user to enter a number, and submit it to s**`trprocess.php`**.

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>Using string functions </title>
</head>
<body>
    <h1>Web Development Form – Lab 3</h1>
    <form action = ______(7)_____ method = ____(8)_____ >
       __(9-10)_____________________________________
       ____________________________________________
       ____________________________________________
       ____________________________________________
       ____________________________________________
```

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

Test in the browser.

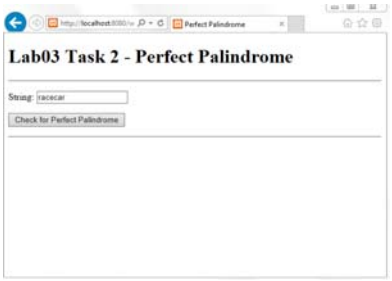
Task 2: Practicing string functions (3 marks)

Background:
A perfect palindrome is a word or phrase that is identical forward or backward, such as the word “racecar”.
A standard palindrome is similar to a perfect palindrome except that spaces and punctuation are ignored.
For example, “Madam, I'm Adam” is a standard palindrome because the characters are identical forward or backward, provided you remove the spaces and punctuation marks.

Step 1:
Create a file **`perfectpalindromeform.html`** that contains a form with a single text box that allows a user to enter a string, and submit it to **`perfectpalindrome.php`**.

Step 2:
Create a file **`perfectpalindrome.php`** with a script that tests whether a word or phrase, entered by a user through the form is a perfect palindrome.

Hint: Use the `strrev()` function to reverse the input word or phrase and then use the `strcmp()` function to compare the original word or phrase with the reversed one.
Suggest also converting the strings to lower case, or upper case, before comparing them.



Task 3: Practicing the use of `regular expressions` (2 marks)

Step 1:
Save copies of the scripts created in Task 2 as **`standardpalindromeform.html`** and **`standardpalindrome.php`**.

Step 2:
Modify the script to check for standard palindromes. For standard palindrome, first remove all the punctuation from the phrase before reversing the word or phrase and comparing it with the original one.

Hint: Use the preg_replace() function to perform a regular expression search and replace. The regular expression pattern should match all non-alphanumeric characters.

View in the browser.

Extra Challenge:

Combine the form **`standardpalindromeform.html`** and processing script **`standardpalindrome.php`**., incorporating the form into a script **`standardpalindromeself.php`**.. i.e. Use a single webpage with the script that displays and also processes the form, i.e. the webpage calls itself.

Test in the browser.

Note:
If you want to prevent any problems that might be caused by users including html markup, such as < or >, as input in forms, then either replace them, or use the functions `htmlspecialchars().`or `htmlentities()` to convert these characters.

See http://php.net/manual/en/function.htmlspecialchars.php
and also see http://www.php.net/manual/en/function.htmlentities.php