**Project 02\_03\_01**

In this Project we will build a series of PHP applications to test our knowledge of the unit. Be sure to document al of your code to indicate your understanding.

 

**Part 01**

1. Create a new file called ***ValidateCreditCard.php***. Scaffold a basic HTML code layout into it. Set the <title> to ***Validate Credit Card***. Make sure to have the ***modernizr*** <script> linked in. Perform a server/browser test.
2. Add an ***<h2>*** element to the body with the content set to ***Validate Credit Card***. Place a set of PHP standard ***script*** ***delimiters*** in the <body> of the document.
3. Declare a ***$creditCard*** array that contains three values, an empty string, a valid credit card with numbers and dashes, and an invalid credit card that has four initial uppercase O’s:  
    ***<?php  
    $creditCard = array("",  
    "8910-1234-5678-6543",  
    "OOOO-9123-4567-0123");  
    ?>***Do a server/browser test for syntax.
4. Add a ***foreach*** loop that iterates through the array testing that each credit card has a value:  
    ***foreach ($creditCard as $indexNumber => $cardNumber) {  
    if (empty($cardNumber)) {  
    echo "<p>Credit Card Number $indexNumber is invalid   
    because it contains an empty string.</p>";  
    }  
    }***Do a server/browser test.
5. Add an ***else*** clause to validate the credit card number, which removes any dashes and spaces in the number:  
    ***else {  
    $creditCardNumber = $cardNumber;  
    $creditCardNumber = str\_replace("-", "",   
    $creditCardNumber);  
    $creditCardNumber = str\_replace(" ", "",   
    $creditCardNumber);  
    echo "<p>Credit Card Number $indexNumber   
    $creditCardNumber removed dashes and   
    spaces.</p>";  
    }***Do a server/browser test.
6. Add a nested ***if…else*** clause to validate that the credit card number, which tests to see if the number is numeric:  
    echo "<p>Credit Card Number $indexNumber   
    $creditCardNumber removed dashes and spaces.</p>";  
    ***if (!is\_numeric($creditCardNumber)) {  
    echo "<p>Credit Card Number $indexNumber   
    $creditCardNumber is invalid because it contains a   
    non-numeric character.</p>";  
    }  
    else {  
    echo "<p>Credit Card Number $indexNumber   
    $creditCardNumber is a valid Credit Card   
    Number.</p>";  
    }***Do a server/browser test.

**Part 02**

1. Create a new file called ***CompareStrings.php***. Scaffold a basic HTML code layout into it. Set the <title> to ***Compare Strings***. Make sure to have the ***modernizr*** <script> linked in. Perform a server/browser test.
2. Add an ***<h2>*** element to the body with the content set to ***Compare Strings*** Place a set of PHP standard ***script*** ***delimiters*** in the <body> of the document.
3. Declare and initialize two string variables, ***$firstString*** and ***$secondString***:  
    ***<?php  
    $firstString = "Geek2Geek";  
    $secondString = "Geezer2Geek";  
    ?>***Do a server/browser test for syntax.
4. Add an ***if…else*** statement to determine if each string has a value:  
    ***if ( !empty($firstString) && !empty($secondString) ) {  
    echo "No strings empty.";  
    }  
    else {  
    echo "<p>Either the \$firstString variable or the   
    \$secondString variable does not contain a value so the   
    two strings cannot be compared.</p>";  
    }***Do a server/browser test.
5. Replace the ***echo*** statement inside the ***if*** clause with the following ***nested if…else*** statement. This will determine if the strings are the same, or how many similarities there are in them and how many must be changed to make them the same:  
    ***if ($firstString == $secondString) {  
    echo "<p>Both strings are the same.</p>";  
    }  
    else {  
    echo "<p> Both strings have " .   
    similar\_text($firstString, $secondString) . "   
    character(s) in common.</p>";  
    echo "<p>You must change " . levenshtein($firstString,   
    $secondString) . " character(s) to make the strings   
    the same.</p>";  
    }***Do a server/browser test.

**Part 03**

1. Create a new file called ***ValidateLocalAddress.php***. Scaffold a basic HTML code layout into it. Set the <title> to ***Validate Local Address***. Make sure to have the ***modernizr*** <script> linked in. Perform a server/browser test.
2. Add an ***<h2>*** element to the body with the content set to ***Validate Local Address***. Place a set of PHP standard ***script*** ***delimiters*** in the <body> of the document.
3. Declare and initialize an array named ***$email***:  
    ***<?php  
    $email = array("jsmith123@example.org",  
    "john.smith.mail@example.org",  
    "john.smith@example.org",  
    "john.smith@example",  
    "jsmith123@mail.example.org");  
    ?>***Do a server/browser test for syntax.
4. Add a ***foreach*** loop that iterates through the array testing that each email has the correct format:  
    ***foreach ($email as $emailAddress) {  
    echo "The email address &ldquo;" . $emailAddress .   
    "&rdquo; ";  
    if (preg\_match("/^(([A-Za-z]+\d+)|" . "([A-Za-z]+\.[A-  
    Za-z]+))" . "@((mail\.)?)example\.org$/i",   
    $emailAddress) == 1) {  
    echo " is a valid e-mail address.<br>";  
    }  
    else {  
    echo " is not a valid e-mail address.<br>";  
    }  
    }***Do a server/browser test.