Individual Assignments

For this individual assignment, please complete all 13 given exercises here. Please note that the yellow highlight on text to print to the browser is intended only to make it easy to see what your goal is. You don't need to highlight the text in your scripts.

1. For your first PHP exercise, create a PHP page, with the standard HTML <head>, <title> and <body> tags. This is not strictly necessary but is good practice and should the the first step for each exercise. Consider creating a template file with these tags already in place. Remember to save your file with the .php extension. Inside the <body> tag, create a PHP section that will show the text "Hello World!" in the browser. Check your work by opening the page in a browser. If you get a blank page, check to be sure that you ended each statement with ";" and that you included the opening and closing PHP markers. When you are ready, move on to the answer script.

2. For your second PHP exercise, echo the following statement to the browser:"Twinkle, Twinkle little star." Next, create two variables, one for the word "Twinkle" and one for the word "star". Echo the statement to the browser, this time substituting the variables for the relevant words. Change the value of each variable to whatever you like, and echo the statement a third time. Remember to include code to show your statements on different lines.

3. PHP includes all the standard arithmetic operators. For this PHP exercise, you will use them along with variables to print equations to the browser. In your script, create the following variables:
   $x=10;
   $y=7;
   Write code to print out the following:
   10 + 7 = 17
   10 - 7 = 3
   10 * 7 = 70
   10 / 7 = 1.4285714285714
   10 % 7 = 3
   Use numbers only in the above variable assignments, not in the echo statements. You will need a third variable as well.

4. Arithmetic-assignment operators perform an arithmetic operation on the variable at the same time as assigning a new value. For this PHP exercise, write a script to reproduce the output below. Manipulate only one variable using no simple arithmetic operators to produce the values given in the statements.
   Hint: In the script each statement ends with "Value is now $variable."
   Value is now 8.
   Add 2. Value is now 10.
   Subtract 4. Value is now 6.
   Multiply by 5. Value is now 30.
   Divide by 3. Value is now 10.
   Increment value by one. Value is now 11.
   Decrement value by one. Value is now 10.

5. When you are writing scripts, you will often need to see exactly what is inside your variables. For this PHP exercise, think of the ways you can do that, then write a script that outputs the following, using the echo statement only for line breaks.
   string(5) "Harry"
   Harry
   int(28)
   NULL

6. For this PHP exercise, write a script using the following variable:
   $around="around"; Single quotes and double quotes don't work the same way in PHP. Using single quotes (' ') and the concatenation operator, echo the following to the browser, using the variable you created:
   What goes around comes around.

7. PHP allows several different types of variables. For this PHP exercise, you will create one variable and assign it different values, then test its type for each value. Write a script using one variable “$whatsit” to print the following to the browser. Your echo statements may include no words except “Value is”. In other words, use the function that will output the variable type to get the requested text. Use simple HTML to print each statement on its own line and add a relevant title to your page. Include line breaks in your code to produce clean, readable HTML.
   Value is string.
   Value is double.
   Value is boolean.
   Value is integer.
   Value is NULL.

8. In this PHP exercise, you will use a conditional statement to determine what gets printed to the browser. Write a script that gets the current month and prints one of the following responses, depending on whether it's August or not:
   It's August, so it's really hot.
   Not August, so at least not in the peak of the heat.
   Hint: the function to get the current month is *'date('F', time())'* for the month's full name.

9. Loops are very useful in creating lists and tables. In this PHP exercise, you will use a loop to create a list of equations for squares.Using a for loop, write a script that will send to the browser a list of squares for the numbers 1-12.
   Use the format, "1 * 1 = 1", and be sure to include code to print each formula on a different line.

10. In the next PHP exercise, you will request input from the user, then move the user's response from one file to another and do something with it.
    Create two separate files. The first will contain a form with one input field asking for the user's favorite city. Use the post method for the form. Although this file contains no PHP code, on my localhost, it needs the .php extension to successfully call the second file. The second file will contain PHP code to process the user's response. (In this case, something very simple.) After the user clicks the submit button, echo back Your favorite city is $city., where $city is the input from the form.
    Hint: the variable that contains the user's input is an array. Arrays will be addressed in future exercises, but this particular array needs to come into play here. The array variable is *$_POST['name']*, where 'name' is the name of your input field.

11. For this PHP exercise, create a form asking the user for input about the weather the user has experienced in a month of the user's choice. In separate text fields, request the city, month and year in question. Below that, show a series of checkboxes using the same weather conditions you used in Arrays Ex. #1. (Those values were: rain, sunshine, clouds, hail, sleet, snow, wind.) Set up the form to create an array from the checked items. In the response section of your script, create an array using the city, month and year the user entered as values. Print the following response "In $city in the month of $month $year, you observed the following weather:", where $city, $month and $year are values from the array you created.

12. For this PHP exercise, you will rewrite the rectangle area function from #2 once again, this time to accept user input. Present a form to the user with the message "Please enter the values of the length and width of your rectangle." Below this, supply two text boxes, one for length and one for width. Using your function to process the user supplied values, return the result statement from the previous exercise to the user. Reminder: the statement was "A rectangle of length $l and width $w has an area of $area.", where $l and $w are the arguments and $area is the result.

13. For this PHP exercise, first create an array called $months. Use the names of the months as keys, and the number of days for each month as values. For February, use the following for your value: "28 days, if leap year 29".
    Next, write a function to create an option element for a form's select field. Make sure each option will be upper case. Both the array and the function should precede the HTML for the page.
    Once again, you will be requesting user input. Create a form for the user with the request, "Please choose a month." Next, provide a select field with the months as options, looping through the array you created and using the function to create the option elements