INFO1-CE9224 Lab Assignments

You're not responsible for turning these assignments in. They are for your own benefit so that you can learn in class and I can help you as you go. Choose whichever assignments you'd like in any order.

If you have any questions, please feel free to ask during the lab.

Exercises

Date Validation

Using php's checkdate() function, print out which dates are valid and which dates are invalid:

Month: 5 Day: 12 Year: 2023
Month: 2 Day: 29 Year: 1996
Month: 2 Day: 29 Year: 2001
Month: 9 Day: 31 Year: 1999
Month: 15 Day: 27 Year: 2012

Using php's strtotime() function, print out which dates are valid and which dates are invalid:

today
two hours from now
February 29, 2012
February 29, 2011
September 31, 2012
+3 Hours
-2 Days
Forever Ago

Using php's date() function, print out the valid dates from the strtotime() exercise is the following formats (ignore the actual date below, it's just an example):

March 23, 2012 HH:MM:SS AM/PM (12 hour time)
03-23-2012 HH:MM:SS (24 hour time)
2012-03-23 HH:MM:SS (24 hour time)

For this last exercise you may have to do a little research: http://php.net/date

Assignment #1

Using the template, http://davehauenstein.com/nyu/INFO1-CE9224-2012-Summer/labs/class6/template.txt, validate each field. Each field should be validated based on the following criteria:

- **username**: alphanumeric and no longer than 15 characters long
- **first_name**: only letters (in reality, there are additional characters that are ok, but ignore this requirement for now).
- **last_name**: only letters (in reality, there are additional characters that are ok, but ignore this requirement for now).
- **email**: Using a regular expression, validate this field. This will require you to read about preg_match() at http://php.net/preg_match. You can write your own regular expression for this by reading ahead in the book, or you can find one online to use.

If the fields are invalid redisplay the form and print a message next to each field that's invalid. Otherwise print the results to the screen and don't display the form again.

Optional requirements:

- If the form is valid, use an HTTP redirect to display the results.
- Write functions to validate each field, taking the value of the field as the first argument and an array as the second argument. The second argument should be passed as reference and should contain the error messages.
- Add a third, optional parameter called $filter, which is a callback. The callback should be expected to take in the value as an argument, and then return the value back. In the meantime, it should run a filter on the value BEFORE the validation takes place. For example, the last name may have single quotes that we want to strip out before validating that it is alpha-numeric.