# Databases and the Web – Exercise 8

This week we'll enhance the data entry page built in exercises 5-7 with some decent validation.

## Outcomes:

When you've finished this exercise you should be able to use regular expressions for data validation.

**Debugging**: Because PHP is *server-side* you **must** check your code in the browser through the web server "http://..." *not* by loading the page as "file://"! It is a good idea to:

i. Open the PHP page in the browser (http://localhost/… or http://studentnet/~kxxxxxxx/...)

ii. Make small changes to code.

iii. Save the file (upload to StudentNet if necessary).

iv. Refresh the page in the browser.

v. Fix any reported syntax errors.

vi. Test the changes.

Repeat *ad nauseum*. (Get used to using Alt-Tab to cycle through windows ☺)

## Task 1: Read and digest the "phpmaster" articles on regular expressions

1. http://phpmaster.com/practicing-regular-expressions/
2. http://phpmaster.com/regular-expressions/
3. ... and have a laugh at http://thedailywtf.com/Articles/The-Clbuttic-Mistake-.aspx

## Task 2: Validate data using regular expressions.

Your city data entry page from exercise 7 should have verified the length of entered data even if you did not use regular expressions to check the *contents*.

1. Following this week's lecture you should know how to build regular expressions into your validation so use them to validate the city data:
   - Use **`preg_match`** in PHP (mandatory server-side) and ***`String`*`.match`** in JavaScript (optional client-side).
   - The anchors **^** and **$** are used to ensure the *whole string* matches.
   - **`\w`** matches word characters (do you need to worry about "_"?), **`\d`** digits.
   - **`{`*`m`*`,`*`n`*`}`** means "between ***m*** and ***n*** characters".
2. Experiment with the HTML5 "required", "pattern", "placeholder" *etc.* attributes – in RateMyJam you should come up with a rationale for using them or not, based on the users' likely requirements and supported browsers.