Form Validation Exercise

This is an exercise in forms and the validation of their inputs. The validation of inputs was traditionally done using JavaScript. Now it is built into HTML5.

The validation of forms is a courtesy extended to users. It should never be regarded as a security precaution for sanitising data. A hacker will be able to write code which will present any input data s/he wants. Data sanitisation should be done at the server end using PHP.

If you are stuck, use W3Schools as a reference (HTML input attributes) or ask for help in Lab.

# What you have to do:

1. Put the two provided files echo.php and form-get.htm into a folder on your web server. As you edit form-get.htm, you can test it with the aid of echo.php.
2. The fields were all created using a simple input tag without attributes. First determine a reasonable maximum length for each field. The default width for an input field is 'size="20"'. That means 20 characters. Change that to a more appropriate length for each field.
3. Make the password field a password field, i.e. the characters appear as blobs, in case someone is reading over your shoulder.
4. Check that the email address really is an email address. Use the "pattern" attribute or use 'type="email"'.
5. Check that the postcode is valid. Use the "pattern" attribute and get the pattern itself from:  
   <http://html5pattern.com> .
6. Check the phone numbers, either by using input 'type="number"', or by using "pattern" and a regular expression.
7. Test your work. If it hasn't been tested, it probably contains bugs.
8. The top 6 fields are required and may not be left blank. Use a "required" attribute in those fields and test that it works.
9. The two passwords must be the same. This is a check that the user did not put an unintentional typo into his/her input. Remember, they can't check by comparing them because they are all blobs. Use JavaScript to check that the passwords are the same as soon as the field for password 2 loses focus. use the HTML trigger event "onchange=". If the passwords don't match, tell the user immediately. Test your work.
10. So far we have deliberately used the GET method. This has the advantage that the output is visible in the URL field of the browser while you are testing. However, the passwords are visible to anyone who cares to take a look. Change the method to POST.