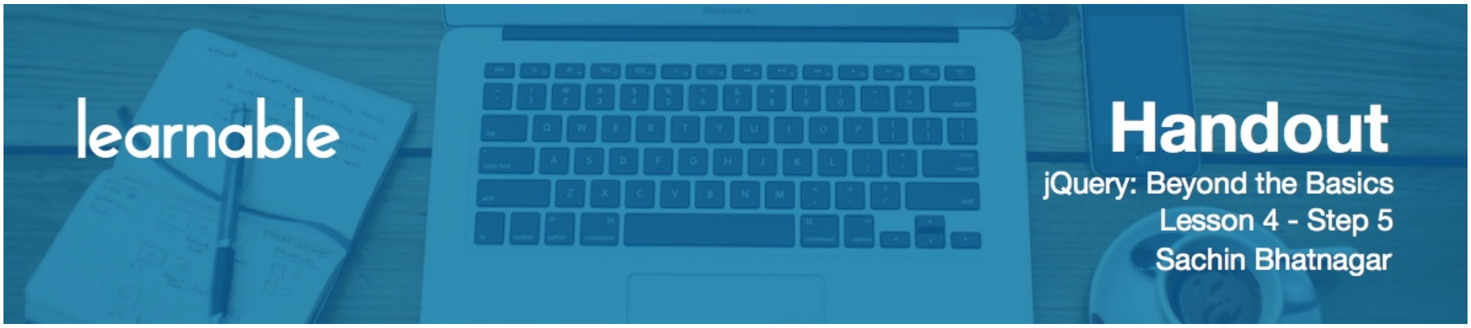

learnable

Handout

jQuery: Beyond the Basics
Lesson 4 - Step 5
Sachin Bhatnagar

Introduction to form validation with jQuery plugins

Forms allow the user to fill in data that can be used for a variety of things. But ensuring format accuracy and validity can be a pretty complex task. Therefore, a form validator implementation is an absolute must for all websites that carry some sort of a form for capturing user input. Things like email addresses, phone numbers, credit card numbers, and passwords require strong validity checks before they can be passed on to the server.

Introduction to parsley.js

For the purposes of form validation, there is an excellent plugin called Parsley.js. One of the strong aspects of Parsley is that it is heavily customizable and open, allowing developers to mold it into any kind of a project.

Validating forms with parsley.js

Open up *formValidator_Begin.htm* to begin. You can also visit the parsleyjs.org to browse documentation and download the plugin itself. For this project, however, we'll employ a CDN:

```
<script src="//cdnjs.cloudflare.com/ajax/libs/parsley.js/2.0.7/parsley.min.js"></script>
```

On to the code:

```
$(function() {
    $("#submitForm").click(function() {
        var validateForm = $("#registerForm").parsley().validate();
        if(validateForm) {
            alert("Your form is ready to be submitted !");
        }
    });
});
```

If you go back to the browser, reload the page and hit "Register", you'll notice that the alert box appears. Peek inside the page using the Inspect element - you'll notice that next to every form field we have a new unordered list with a class name of `parsley-errors-list`. This element is being inserted into the DOM by the Parsley plugin, and this'll be used to display the errors.

Parsley invites you to style this element as you please. For this example, I've added a set of styles in the *formvalidator.css*.

We now have to define validation rules:

```
<div class="container">
  <form id="registerForm">
    <div class="formHolder">
      <div class="labelHolder">
        <label>Full Name</label>
      </div>
      <div class="fieldHolder">
        <input type="text" name="fullname" autocomplete="off" required/>
      </div>
    </div>
    <div class="formHolder">
      <div class="labelHolder">
        <label>E-Mail</label>
      </div>
      <div class="fieldHolder">
        <input type="email" name="email" autocomplete="off" required/>
      </div>
    </div>
    <div class="formHolder">
      <div class="labelHolder">
        <label>Phone</label>
      </div>
      <div class="fieldHolder">
        <input type="phone" name="text" autocomplete="off" pattern="/^\d{10}$/" required/>
      </div>
    </div>
    <div class="formHolder inflated">
      <div class="labelHolder">
        <label>Postal Addr.</label>
      </div>
      <div class="fieldHolder">
        <textarea name="postal_address" required></textarea>
      </div>
    </div>
    <div class="formHolder pushdown01">
      <div class="labelHolder">
        <label>Password</label>
      </div>
      <div class="fieldHolder">
        <input type="password" name="password" data-parsley-type="alphanum" data-parsley-length="[8,12]" required/>
      </div>
    </div>
    <div class="formHolder">
      <div class="labelHolder">
        <label>Confirm Password</label>
      </div>
      <div class="fieldHolder">
        <input type="password" name="confirm_password" data-parsley-equalto="input[name='password']" required/>
      </div>
    </div>
    <div class="formHolder">
      <a href="#" id="submitForm">Register</a>
    </div>
  </form>
</div>
```

The `required` attribute means that this field has to be filled before the form can be submitted.

The `pattern` attribute accepts a regular expression to validate the input.

`data-parsley-type` attribute means that the value in this field should be of specific type (alphanumeric in this case).

`data-parsley-Length` set to `[8,12]` means that the number of characters for the password must lie between 8 and 12.

For the `confirm_password` field, its contents need to match up to the password input field to be validated, so we set `data-parsley-equalto` and pass in a selector to the password input field.

Check that in your browser!