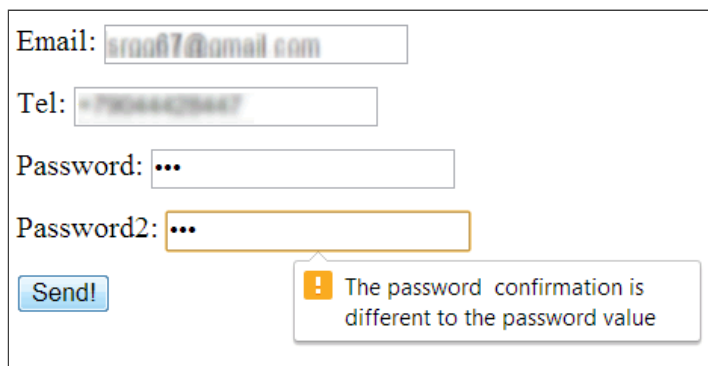# The HTML5 JavaScript form validation API

There is a JavaScript API for form validation. This API will let you use your own validation algorithm (i.e. check that you entered the same password in two different input fields), and customize error messages. Also, together with some HTML/CSS/JavaScript you will be able to make your own message bubbles.
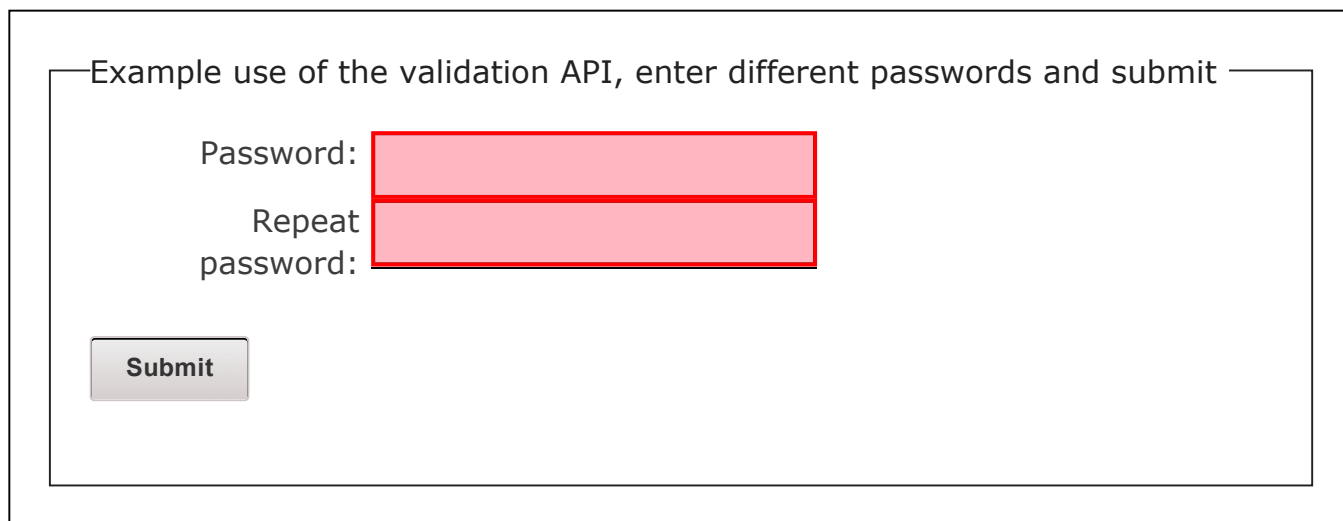
Email: sroo67@gmail.com

Tel: +7904425847

Password: •••

Password2: •••

Send!

⚠ The password confirmation is different to the password value

TYPICAL USE

Example of password checking at JS Bin, be careful to try this example in JS Bin standalone mode (click the small black arrow on the top right of the output tab).

Or you may try it here in your browser:

Example use of the validation API, enter different passwords and submit

Password:

Repeat password:

Submit

Extract from source code:

```html
<html>
<head>
    <title>Example of using the validation API</title>
    <style>
        .myForm input:invalid { background-color:lightPink;}
```

```
        .myForm input:valid { background-color:lightGreen; }
        .myForm input:required {border: 2px solid red;}
        .myForm input:optional {border: 2px solid green;}
        .myForm label { display: inline-block; width: 140px;text-align: right; }
10.    </style>
    </head>
    <body>
    <form class="myForm">
      <fieldset>
        <legend>Example use of the validation API</legend>
        <label for="password1" >Password:</label>

     <input type="password" id="password1"oninput="checkPasswords();" required>
        <p>
        <label for="password2">Repeat password:</label>

     <input type="password" id="password2"oninput="checkPasswords();" required>
        <p>
22.      <button>Submit</button>
      </fieldset>
    </form>

    <script>
     function checkPasswords() {
        var password1 =document.getElementById('password1');
        var password2 =document.getElementById('password2');
        if (password1.value != password2.value) {
          password2.setCustomValidity('Passwords do not match!');
33.      } else {
          password2.setCustomValidity('');
        }
      }
    </script>
    </body>
    </html>
```

Explanations: the validity API proposes a `setCustomValidity()` method available on input DOM objects. This method allows you to customize error messages. It takes a string parameter. When this string is empty, the element is considered *valid*, when the string is not empty, the field is *invalid* and the validation error message displayed in the bubble will

be equal to that string.

At lines 17 and 20 we added an `input` event listener: each time a key is typed, the `checkPasswords()` function is called.

Lines 28 and 29 get the input fields' values, and lines 31-35 check if the passwords are the same and set the validity of the field using the validation API's method `setCustomValidity(error_message)`.