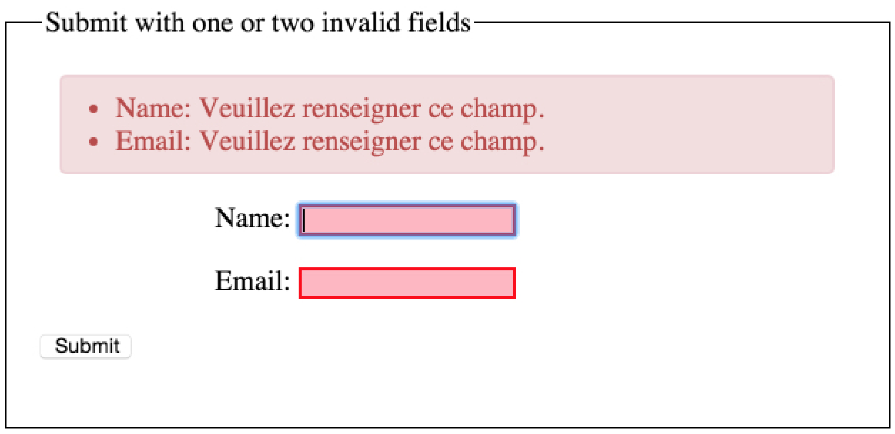
Custom validation: changing the default behavior, aggregating error messages, removing bubbles, etc.


Submit with one or two invalid fields

- Name: Veuillez renseigner ce champ.
- Email: Veuillez renseigner ce champ.

CRITICISM OF THE DEFAULT BEHAVIOR OF HTML5 BUILT-IN VALIDATION

The techniques we have seen so far for enhancing HTML forms are powerful and provide interesting features, but are also criticized by Web developers:

- Browser support is still not 100% complete (Safari and Internet Explorer still lack several important features),
- It is not possible to aggregate error messages. On submission, browsers show an error bubble next to the first invalid field, and there is no built-in way to *display all error messages for all invalid fields at the same time*,
- You cannot style the bubbles.

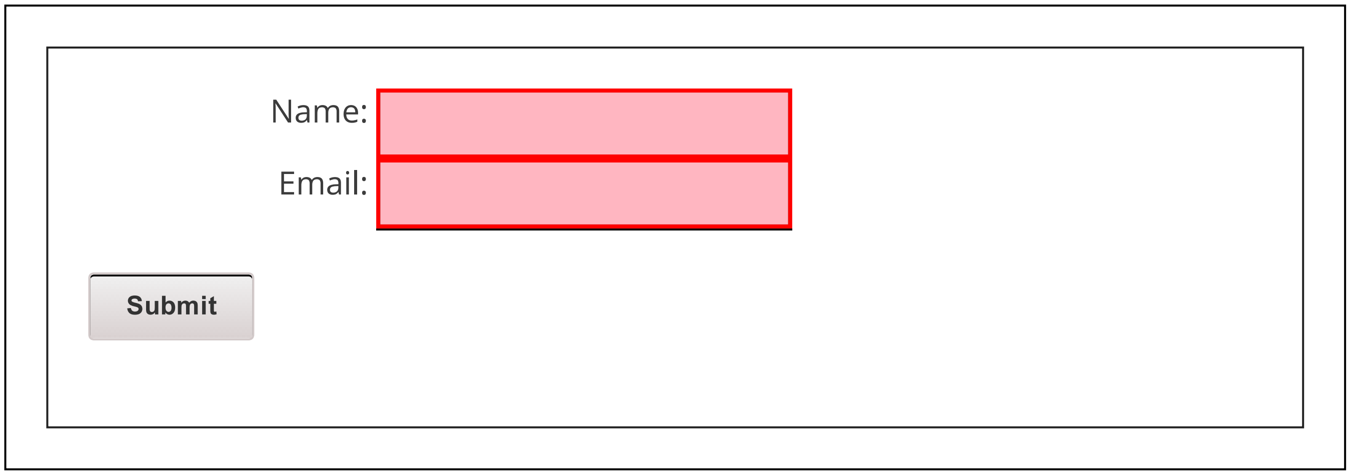
However, the validation API gives enough power to make your own validation behavior, overriding the default when necessary.

Here is an adaptation of work presented at the developer.telerik.com Web site. This link is really worth reading, as it presents different approaches and gives external references for those who would like to go further.

EXAMPLE THAT SHOWS AGGREGATION OF ERROR MESSAGES +

OVERRIDING DEFAULT BEHAVIOR

Try the online example at JS Bin, or try it here in your browser: enter invalid values and submit with one or two invalid fields.



Complete source code:

```
<!DOCTYPE html>
<html>
 <head>
   <meta charset="utf-8">
   <title>Aggregating error messages</title>
   <style>
       input:invalid { background-color:lightPink;}
       input:valid { background-color:lightGreen; }
       input:required {border: 2px solid red;}
       input:optional {border: 2px solid green;}

       .error-messages {
           display: none;
           margin: 0 10px 15px 10px;
           padding: 8px 35px 8px 30px;
           color: #B94A48;
           background-color: #F2DEDE;
           border: 2px solid #EED3D7;
           border-radius: 4px;
       }
       fieldset {
          border:1px solid;
          padding:20px;
```

```
      }
   </style>
</head>
<body>
<form>
    <fieldset>
        <legend>Submit with one or two invalid fields</legend>
        <ul class="error-messages"></ul>

        <label for="name">Name:</label>
        <input id="name" name="name"required>
        <p>
        <label for="email">Email:</label>
        <input id="email" name="email"type="email" required>
        <p>
        <button>Submit</button>
    </fieldset>
</form>

<script>
   function replaceValidationUI(form) {
      // Suppress the default bubbles
         form.addEventListener("invalid", function(event) {
         event.preventDefault();
      }, true);

      // Support Safari, iOS Safari, and the Android browser —
each of which
      // do not prevent form submissions by default
      form.addEventListener("submit",function (event) {
         if (!this.checkValidity()) {
            event.preventDefault();
         }
      });

      // Container that holds error messages. By default it has
a CSS
      // display:none property
      var errorMessages =form.querySelector(".error-messages");


var submitButton =form.querySelector("button:not([type=button]),
input[type=submit]");

       submitButton.addEventListener("click",function (event) {

 var invalidFields =form.querySelectorAll("input:invalid");
           var listHtml = "";
           var
errorMessagesContainer =form.querySelector(".error-messages");
           var label;

           // Get the labels' values of their name attributes +
the validation error
           // message of the corresponding input field using the
validationMessage
           // property of input fields
           // We build a list of <li>...</li> that we add to the
error message container
           for (var i = 0; i <invalidFields.length; i++) {

 label =form.querySelector("label[for=" +invalidFields[ i ].id + "]");
               listHtml += "<li>" +
                           label.innerHTML +
                           " " +
                           invalidFields[i ].validationMessage +
                           "</li>";
           }

           // Update the list with the new error messages
           errorMessagesContainer.innerHTML =listHtml;

           // If there are errors, give focus to the first
invalid field and show
           // the error messages container by setting its CSS
property display=block
           if (invalidFields.length > 0){
              invalidFields[ 0 ].focus();
              errorMessagesContainer.style.display ="block";
           }
       });
   }
   // Replace the validation UI for all forms
   var forms =document.querySelectorAll("form");
   for (var i = 0; i < forms.length; i++){
       replaceValidationUI(forms[ i ]);
   }
 </script>
104. </body>
</html>
```

Explanations:

- *Line 32*: we added an empty unnumbered list (`<ul>..</ul>`) to the form, with the CSS `class="error-messages"`. We will use this class attribute for styling, and hiding by default, the error messages using CSS (see lines 12-20, line 13 hides the messages by default).
- *Lines 97-102* look at all forms in the document and call a function that will replace the default validation behavior for all of them: the `replaceValidationUI(form)` function.
- This function first disables all default behavior (no more display of bubbles during form submission), this is done at*lines 45-57*.
- *Line 66*: we add a `click` listener to the submit button of the current form.
- *Line 67* gets all invalid input fields for that form,
- *Lines 76-83*: For each invalid field, we get the value of the`name` attribute of the corresponding label, we also get the validation error message, and we build a list item `(<li>...</li>`).
- *Line 86*: Then we add this list element (a formatted error message corresponding to an invalid input field) to the error message container.
- *Lines 90-93*: The focus is given to the first invalid field that shows an error message.