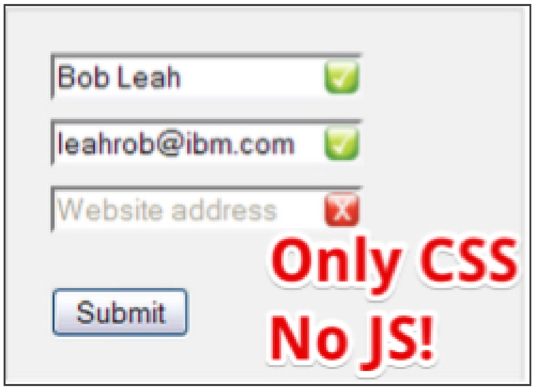
CSS3 pseudo classes bring automatic visual feedback while typing

Most modern browsers propose default behavior for validating input fields and forms.



Bob Leah

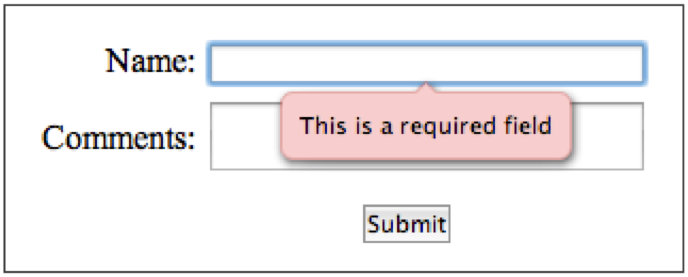
leahrob@ibm.com

The built-in validation system that comes with HTML5 adds automatically a CSS pseudo class to all input fields. Invalid fields (i.e. a badly worded email address in an `<input type="email">` input field), will inherit the `:invalid` pseudo class, valid fields will inherit the `:valid` pseudo class.

A first step to improve your HTML form is to add some CSS rules to your input fields. This will add visual feedback to the validity of input fields values - while the user is typing - like changing the color of the border of input fields, or green/red icons on the right of the field, as shown in the small picture at the top right of this page.

Also, at the time of submitting the form, some extra messages may be displayed as popped up text bubbles.

The default bubble message and visual feedback differ from one implementation to another, but it is possible to customize them, with some limitations that will be explained later.



Name:

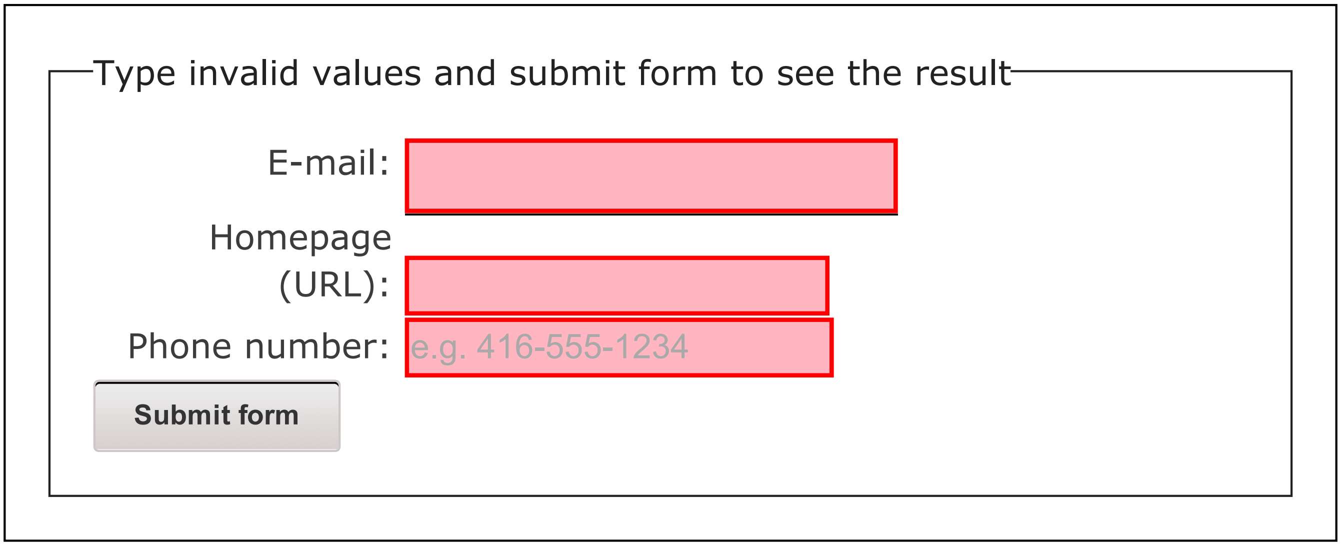
Comments:

This is a required field

For example, Firefox 15+ provides default feedback on the input fields' border (red = invalid, green = ok). This default behavior can be overridden by CSS rules as illustrated in the section about new input type attributes.

EXAMPLE 1: STYLING "REQUIRED", "VALID" AND" INVALID" FIELDS USING CSS3 PSEUDO-CLASSES

Here is an online example at JS Bin, or try it below in your browser:



Type invalid values and submit form to see the result

E-mail:

Homepage (URL):

Phone number: e.g. 416-555-1234

Source code extract:

```
<html>
 <head>
   <title>CSS3 pseudo-classes for form validation visual feedback</title>
   <style>
       input:invalid { background-color: lightPink;}
       input:valid { background-color:lightGreen; }
       input:required {border: 2px solid red;}
       input:optional {border: 2px solid green;}
       fieldset {
          border:1px solid;
          padding:20px;
       }
       .formLabel { display: inline-block; width: 140px;text-align: right; }
   </style>
 </head>
<body>
<form>
```

```
 <fieldset>
   <legend>Type invalid values and see the result</legend>
   <label for="myEmail" class="formLabel">E-mail:</label>
   <input type="email" id="myEmail" required/><br>
   <label for="myURL" class="formLabel">Homepage (URL):</label>
   <input type="url" id="myURL" required/><br>
   <label for="myPhone" class="formLabel">Phone number:</label>
   <input type="phone" id="myPhone"
         pattern="[0-9]{3}-?[0-9]{3}-?[0-9]{4}"
         placeholder="e.g. 416-555-1234" required/>
   <br>
   <button>Submit form</button><br />
 </fieldset>
</form>
```

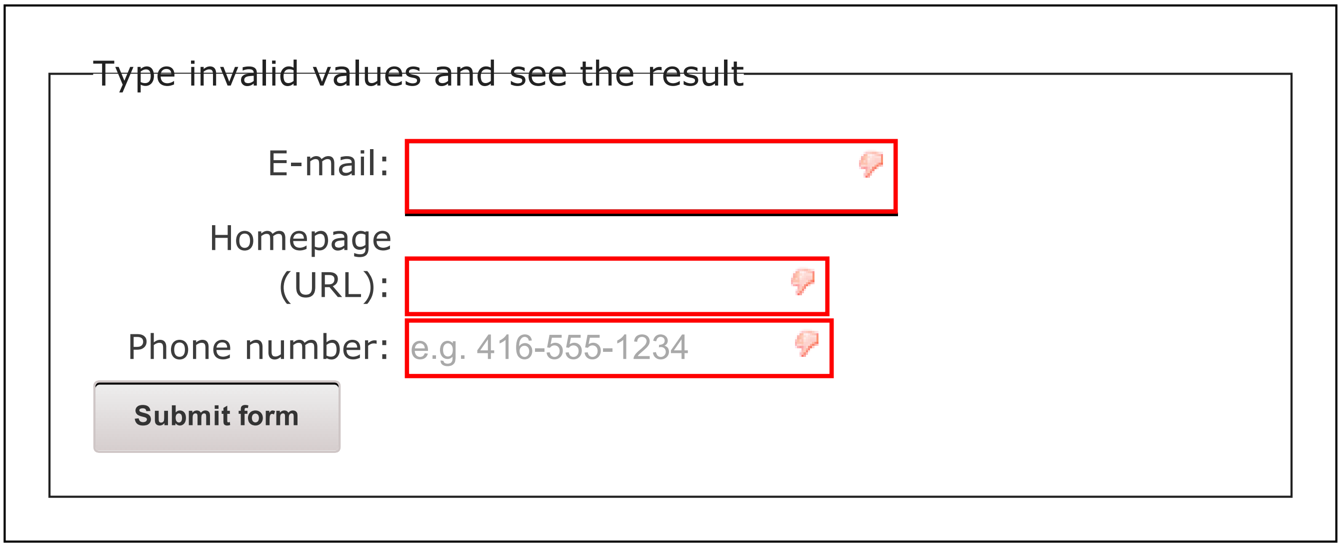
Try the online example with different Web browsers, try it with and without the CSS rules. See the differences between FireFox/Chrome/Opera in the default visual feedback behavior. Do not worry, as all default behavior can be overridden if you provide your own CSS rules.

> **Best practice: except FireFox, no other browser provides default CSS styling for**
> **valid/invalid/optional input fields. We recommend you to ALWAYS provide default CSS rules that give visual feedback to user's input.**

Safari and old browsers will still need a polyfill as of June 2015 (see the different polyfills at the end of this page).

NICE TRICK: ADD CSS TRANSITIONS + AN ICON/MARKER ON THE RIGHT OF INPUT FIELDS

Try this online example at JS Bin or try it here in your browser. This example adds a small icon that changes depending on the validity of the input field:



Type invalid values and see the result

E-mail:

Homepage (URL):

Phone number: e.g. 416-555-1234

Source code extract:

```
.myForm input:focus {
  padding-right:70px;
}
 .myForm input {
  transition: padding .25s;
}
.myForm input:required:valid {
  background:url('http://i.imgur.com/BJolppS.png')no-repeat right top;
}
11. .myForm input:required {
  background:url('http://i.imgur.com/7pIN7wz.png')no-repeat right top;
}
```

This time, we just added an attribute `class="myForm"` to our form, in order not to interfere with the other examples in this page, and we tweaked the CSS rules a little.

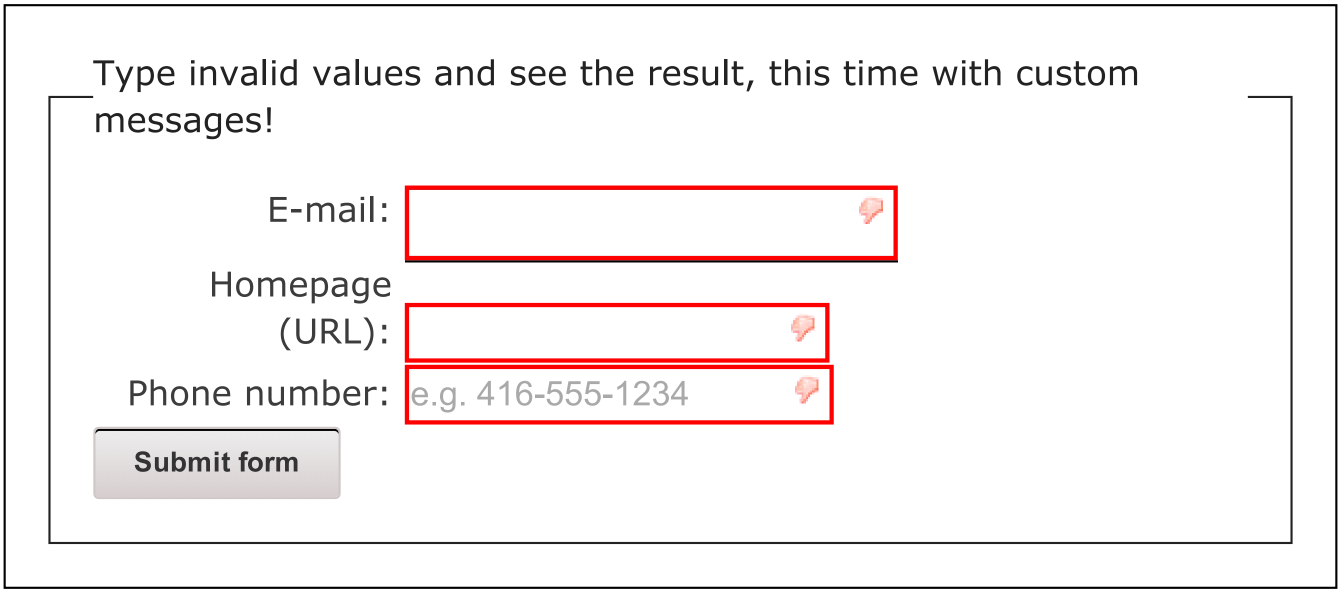
The rule at line 1 says that any time we click on an input field, it will enlarge itself to the right, while the rule at line 4 will make this animated.

The rules at lines 8 and 11 target the input fields with a`required` attribute. They will change the background by displaying a small green or red icon, corresponding to the valid/invalid status of the input field.

USE THE `TITLE` ATTRIBUTE FOR SPECIFYING A CUSTOM MESSAGE

You can simply use the input's `title` attribute to provide a message for pattern-mismatches, and more generally for all validation errors. This solution is really nice and without any JavaScript!

Try the online example at JS Bin, or try it here in your browser (type invalid values and look at the custom messages):



Type invalid values and see the result, this time with custom messages!

E-mail:

Homepage (URL):

Phone number: e.g. 416-555-1234

Extract from source code:

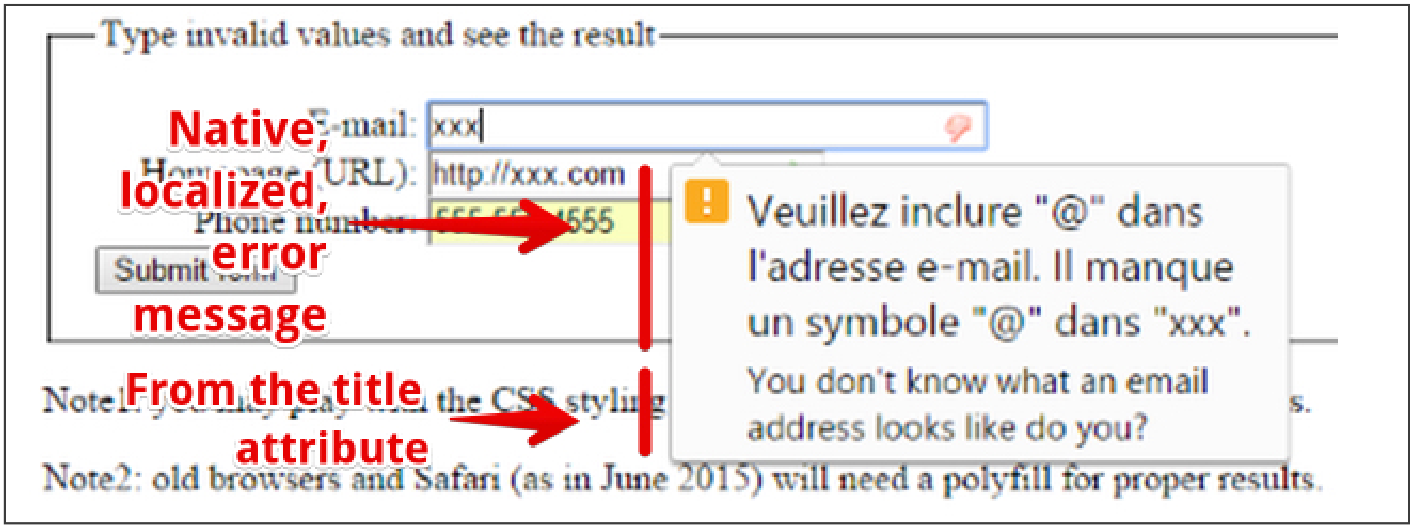
```
<form class="myForm">
  <fieldset>
    <legend>Type invalid values and see the result</legend>
    <label for="myEmail" class="formLabel">E-mail:</label>
    <input type="email" id="myEmail"
          title="You don't know what an email address looks like,
do you?"
          required/><br>
```

```
...
    <button>Submit form</button><br />
  </fieldset>
</form>
```

Beware: browser implementations differ. Chrome, IE, Opera (as of June 2015) will display the `title` attribute value in error message bubbles when the form is submitted, while Safari and FireFox (desktop and mobile) will simply ignore it.

You must also take care of the different languages, otherwise you will get error message bubbles that show some parts in the local language, and the message from the title attribute "as is".

Google Chrome on a French computer:



Type invalid values and see the result

E-mail: xxx

Home page (URL): http://xxx.com

Phone number: 555 55 555

Submit form

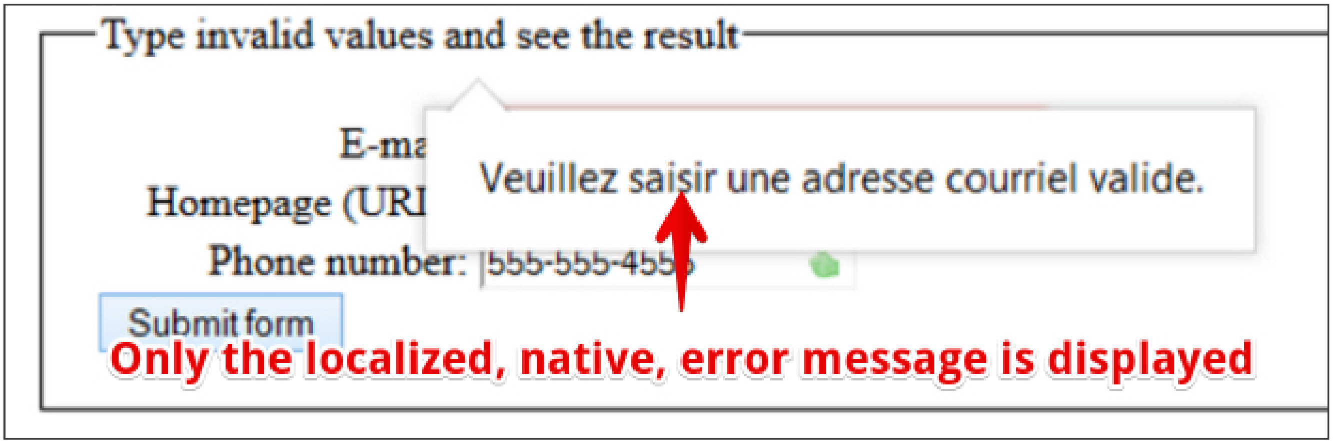
Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "xxx".

You don't know what an email address looks like do you?

Note1: you may play with the CSS styling ... s.

Note2: old browsers and Safari (as in June 2015) will need a polyfill for proper results.

Same example on FireFox, the `title` attribute is ignored:


Type invalid values and see the result
E-ma
Homepage (URI
Phone number: 555-555-4555
Submit form

Veuillez saisir une adresse courriel valide.

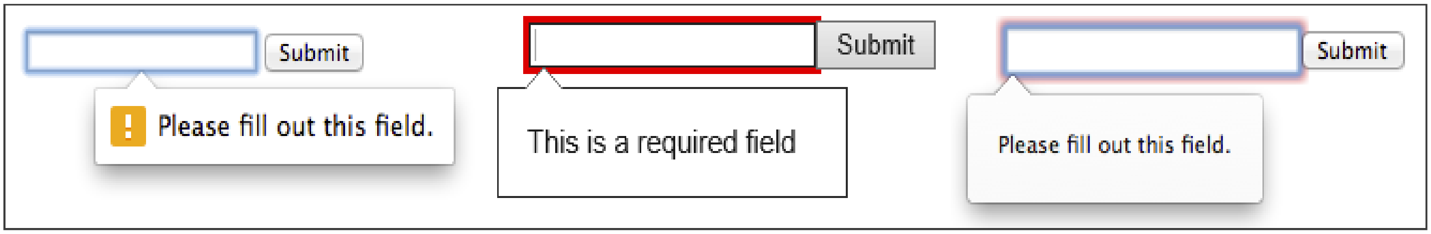
Only the localized, native, error message is displayed

Today, the built-in validation system is an improvement on what existed before HTML5 (aka nothing), but additional work is required if you want fully localized, hand-made validation feedback.

We will show solutions in the last section of this week's course.

HOW TO STYLE THE VALIDATION MESSAGES DISPLAYED IN THE BUBBLES?

Try the examples from this page on different browsers. You may notice differences (from left to right: Chrome, FireFox and IE):



Please fill out this field.

This is a required field

Please fill out this field.

How can we change the style of these bubbles (for example for unifying the look and feel on all browsers)? Unfortunately we can’t.

Chrome used to provide a series of vendor prefixed pseudo-elements (::-webkit-validation-bubble-*), but they were removed with Chrome 28 and later versions. However, there are possibilities that involve the use of the HTML5 form JavaScript API. **At the end of this section, we propose solutions that work**

well.

EXTERNAL RESOURCES (NOT MANDATORY READING)

For those really interested in HTML5 forms, we recommend following the tutorial below, which shows how to give a super look'n'feel to an HTML form, as well as visual feedback using HTML5 new attributes, CSS pseudo elements and form validation API. This tutorial also gives some nice JavaScript tricks and explains HTML4/CSS2 classic enhancements:

- Bring Your Forms Up to Date With CSS3 and HTML5 Validation, tutorial from the http://webdesign.tutsplus.com Web site.

However beware! While this tutorial shows very good tricks about the look'n'feel, it unfortunately shows also a very common mistake:

- The "`for`" attribute of a `<label>` has to match the "`id`" attribute of the related input, not the "`name`" attribute!

Otherwise, it has good content.

POLYFILL FOR BROWSERS THAT DO NOT FULLY SUPPORT THE NEW ATTRIBUTES REQUIRED, PATTERN, ETC. (THIS INCLUDES THE CURRENT SAFARI VERSION)

We recommend using the modernizr.com library together with some external polyfills. These articles are worth reading for those of you who will need retro-compatibility:

- Cross Browser Styling of HTML5 Forms — Even In Older Browsers
- Creating Cross Browser HTML5 Forms Now, Using modernizr, webforms2 and html5Forms
- There is also a small JavaScript based polyfill that brings compatibility to Safari, just for the missing attributes. Notice that a new version is on the way (read the message at the end of the GitHub page of the project).