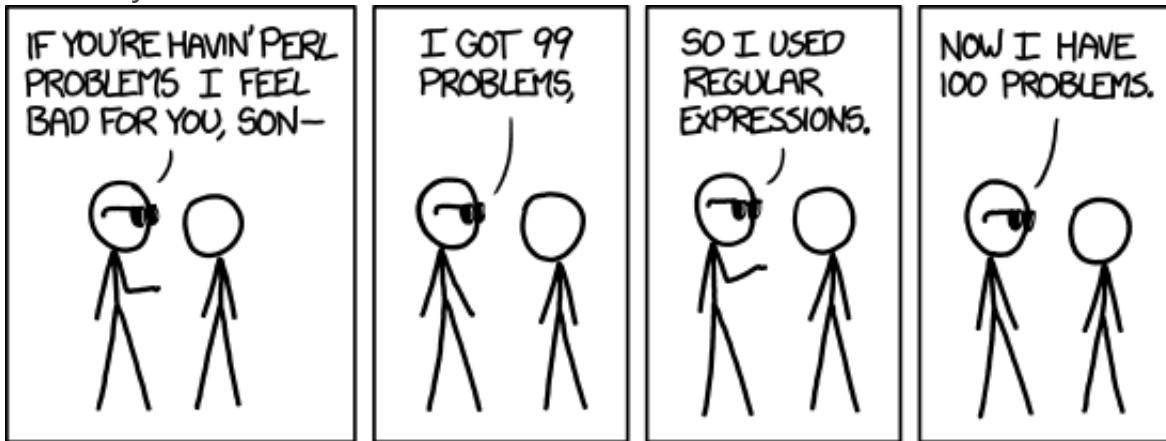


The `pattern` attribute

Cartoon from [xkcd #1171: Perl Problems](#)



The `pattern` attribute enables the validation of the user's input on the fly (also at submission time), based on **regular expressions**. It applies to the `text`, `search`, `url`, `tel`, `email`, and `password` input types.

The `pattern` attribute follows the syntax of [JavaScript regular expressions](#).

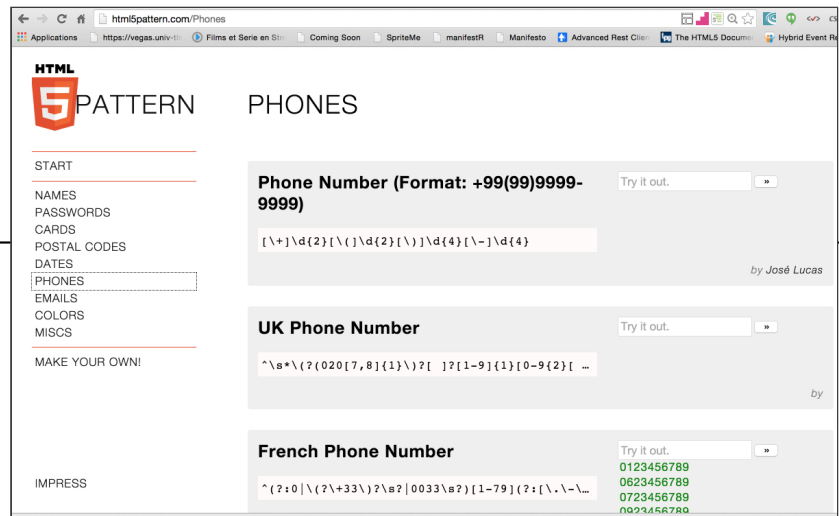
MUST READ: a good catalog of ready-to-go patterns is available at html5pattern.com, an excellent Web site that proposes plenty of JavaScript patterns for the `pattern` attribute of HTML5 forms. The left hand menu proposes categorized patterns for postal codes, phones, URLs for popular sites, etc.

TYPICAL USE

Just add a `pattern` attribute with a value that is the JavaScript regular expression that must match the entire string entered in the field. Note that the empty string is valid by default (except if the `required` attribute is used - this makes empty fields invalid).

It's best practice to systematically add a `title` attribute with a value that

indicates what constitutes a **valid entry**. More on this in the section of this course dedicated to form validation.



```
<input type="text" name="country_code"
       pattern="[A-Za-z]{3}"
       title="3 letter country code"
/>
```

[Online example at JS Bin](#)

Or try it directly in your browser here:

Please enter a 3 letter country code (green = valid, pink = invalid):

With the previous example, until the value of the input field is equal to 3 alphabetic characters, the field is invalid.

As seen in the previous examples, we used some CSS pseudo classes for automatically setting the background-color of the input field as we type.

Complete source code:

```
<!DOCTYPE html>
<html>
<head>
```

```

10. <meta charset="utf-8">
    <title>Example of the pattern attribute</title>
    <style>
        input:invalid {
            background-color: lightPink;
        }
        input:valid {
            background-color: lightGreen;
        }
    </style>
</head>
<body>
    <label for="code">Please enter a 3 letter country code:
</label>
    <input type="text" name="country_code"
        pattern="[A-Za-z]{3}"
        title="3 letter country code"
20.     id="code"/>
</body>
</html>

```

ANOTHER EXAMPLE THAT MIXES SEVERAL OTHER ATTRIBUTES WITH THE PATTERNATTRIBUTE

Online at: <http://jsbin.com/foviki/3/edit>

Try it in your browser:

Attributes used: placeholder (for displaying a ghost example value), pattern, required (empty field = invalid)...

Enter a pseudo (6-12 characters):

Complete source code:

```

<!DOCTYPE html>
<html>
<head>
  <title>Example of use of new HTML5 input field
attributes</title>
  <style>
    input:focus:invalid { background-color: lightPink;}
    input:valid { background-color:lightGreen; }
    input:required {border: 2px solid red; }
    input:optional {border: 2px solid green; }
10.  </style>
  </head>
  <body>
    <p>Attributes used: placeholder (for displaying a ghost
example value), pattern, required (empty = invalid)...
    <p>
    <label for="inputID">Enter a pseudo (6-12
characters): </label>
    <input id="inputID" name="Name"
      placeholder="Name"
      pattern="\w{6,12}"
      required
      title="6-12 characters allowed please"
21.  type="text" />
    </body>
  </html>

```

ANOTHER EXAMPLE THAT USES AN `<INPUT TYPE="URL">` ELEMENT WITH A `PATTERN` ATTRIBUTE ALLOWING ONLY CERTAIN PROTOCOLS

[Online example at JS Bin](#)

Try it in your browser:

Enter the URL of your repository (http, https or ftp):

Source code extract:

```
<input
  id="website"
  name="url"
  type="url"
  placeholder="http://www.domain.com"
  title="http, https or ftp allowed"
  pattern="(http|https|ftp)\:\/\:\/\/[a-zA-Z0-9\-\.\ \/]*"
/>
```

KNOWLEDGE CHECK 5.5.6 (NOT GRADED)

```
<label for="name">Please enter value: </label>
<input type="text" id="name" pattern="[a-zA-Z0-9]+">
```

What kind of values are allowed in this input field (if you wish, get help by visiting the html5pattern.com Web site)?

- ☐ Alphanumeric, no constraint on the length.
- ☐ A lowercase char followed by an uppercase char followed by "0", followed by "-", followed by "9".
- ☐ A string between 0 and 9 characters long (lowercase or uppercase).

