Validation Attributes

In this lesson, we will cover some very useful HTML attributes that will help us perform automatic validation. Let's begin!

HTML5 extended the `<input>` tag with new attributes that specify different kinds of automatic validation. With the required attribute you can specify that an input field must be filled out before submitting the form. When you work with text input, use `maxlength` to define the maximum number of characters allowed in the field.

In the case of numeric values (number, range, and date related input fields) you can specify an acceptable range with the `min` and `max` attributes, and optionally define the legal number intervals with the step attribute. For example, the following markup defines a numeric field that can have only the -3, 1, 5, and 9 values:

```
<input type="number" min="-3" max="9" step="4">
```

In Listing 4-7 of the previous lesson, the Membership ID field's placeholder with the value of "ABC-123" suggested a special format. You can use the pattern attribute to specify a regular expression that the field's value is checked against. For example, to validate Membership ID, you should add the following attribute to the corresponding `<input>` markup:

```
pattern="[A-Za-z]{3}-\d{3}"
```

Before submitting a form, the browser checks if all fields are valid. If not, it changes the appearance of invalid fields and provides some information

about what the source of invalidity is. In most cases, you can use the title

attribute to provide extra information about the expected value and format of
the field.

> 📄 **NOTE:** Different browsers may provide different ways to indicate
> invalid content.

Listing 4-8 provides an example that demonstrates these validation attributes.

Listing 4-8: Using attributes that specify automatic validation

```
<!DOCTYPE html>

<html>
<head>
  <title>Using placeholder</title>
  <style>
    ::-webkit-input-placeholder {
      font-style: italic;
    }

    :-moz-placeholder { /* Firefox 18- */
      font-style: italic;
    }

    ::-moz-placeholder { /* Firefox 19+ */
      font-style: italic;
    }

    :-ms-input-placeholder {
      font-style: italic;
    }
  </style>
</head>
<body>
  <h2>Specify your booking info</h2>
  <form>
    <label>
      First name:
      <input type="text"
        placeholder="John"
        maxlength="20"
        />
    </label>
    <br />
    <label>
      Last name*:
      <input type="text"
        placeholder="Smith"
```
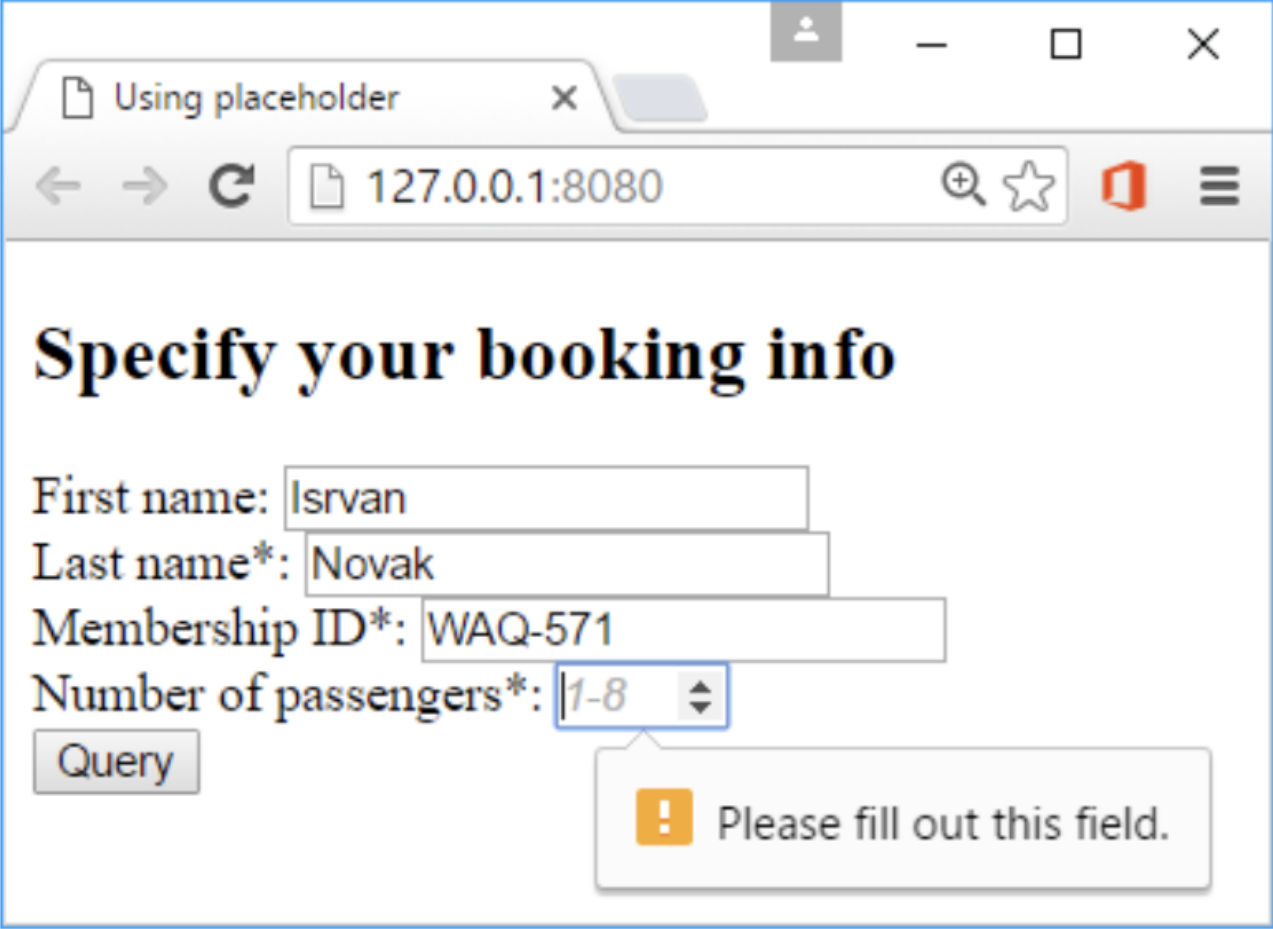
```
            title="Please, specify the last name"
            required />
      </label>

      <br />
      <label>
        Membership ID*:
        <input type="text"
          placeholder="ABC-123"
          required
          title="Three letters, dash, three digits"
          pattern="[A-Za-z]{3}-\d{3}" />
      </label>
      <br />
      <label>
        Number of passengers*:
        <input type="number"
              placeholder="1-8"
              required
              min="1"
              max="8" />
      </label>
      <br />
      <input type="submit" value="Query" />
    </form>
  </body>
</html>
```

As you can see from the markup, all fields except First name are required. Last name and Membership ID define a helping title that is displayed when the field is found invalid. The Number of passengers field expects a numeric value between 1 and 8. Browsers display this form differently.

For example, Edge renders the last field as a simple textbox...

...while Google Chrome provides a numeric control with up and down arrows:

The form in Google Chrome

Great, now that we have covered some very useful validation attributes, let's move on to study validation from a JavaScript perspective in the *next lesson*.

See you there!