Chapter 3 Introduction to HTML5: Part 2

Internet & World Wide Web How to Program, 5/e

OBJECTIVES

In this chapter you'll:

- Build a form using the new HTML5 **`input`** types.
- Specify an **`input`** element in a form as the one that should receive the focus by default.
- Use self-validating **`input`** elements.
- Specify temporary **`placeholder`** text in various **`input`** elements
- Use **`autocomplete input`** elements that help users re-enter text that they've previously entered in a form.
- Use a **`datalist`** to specify a list of values that can be entered in an **`input`** element and to autocomplete entries as the user types.
- Use HTML5's new page-structure elements to delineate parts of a page, including headers, sections, figures, articles, footers and more.

3.1 Introduction

3.2 New HTML5 Form `input` Types

- 3.2.1 `input` Type `color`
- 3.2.2 `input` Type `date`
- 3.2.3 `input` Type `datetime`
- 3.2.4 `input` Type `datetime-local`
- 3.2.5 `input` Type `email`
- 3.2.6 `input` Type `month`
- 3.2.7 `input` Type `number`
- 3.2.8 `input` Type `range`
- 3.2.9 `input` Type `search`
- 3.2.10 `input` Type `tel`
- 3.2.11 `input` Type `time`
- 3.2.12 `input` Type `url`
- 3.2.13 `input` Type `week`

3.3 `input` and `datalist` Elements and `autocomplete` Attribute

3.3.1 `input` Element `autocomplete` Attribute

3.3.2 `datalist` Element

3.3 Page-Structure Elements

3.4.1 `header` Element

3.4.2 `nav` Element

3.4.3 `figure` Element and `figcaption` Element

3.4.4 `article` Element

3.4.5 `summary` Element and `details` Element

3.4.6 `section` Element

3.4.7 `aside` Element

3.4.8 `meter` Element

3.4.9 `footer` Element

3.4.10 Text-Level Semantics: `mark` Element and `wbr` Element

3.1 New HTML5 Form `input` Types

- Figure 3.1 demonstrates HTML5's new form `input` types.
- These are not yet universally supported by all browsers.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 3.1: newforminputtypes.html -->
4  <!-- New HTML5 form input types and attributes. -->
5  <html>
6     <head>
7        <meta charset="utf-8">
8        <title>New HTML5 Input Types</title>
9     </head>
10
11    <body>
12       <h1>New HTML5 Input Types Demo</h1>
13       <p>This form demonstrates the new HTML5 input types
14          and the placeholder, required and autofocus attributes.
15       </p>
16
17       <form method = "post" action = "http://www.deitel.com">
18          <p>
19             <label>Color:
20                <input type = "color" autofocus />
21                   (Hexadecimal code such as #ADD8E6)
22             </label>
23          </p>
```

Fig. 3.1 | New HTML5 form input types and attributes. (Part 1 of 5.)

```
24          <p>
25             <label>Date:
26                <input type = "date" />
27                   (yyyy-mm-dd)
28             </label>
29          </p>
30          <p>
31             <label>Datetime:
32                <input type = "datetime" />
33                   (yyyy-mm-ddThh:mm+ff:gg, such as 2012-01-27T03:15)
34             </label>
35          </p>
36          <p>
37             <label>Datetime-local:
38                <input type = "datetime-local" />
39                   (yyyy-mm-ddThh:mm, such as 2012-01-27T03:15)
40             </label>
41          </p>
42          <p>
43             <label>Email:
44                <input type = "email" placeholder = "name@domain.com"
45                   required /> (name@domain.com)
46             </label>
47          </p>
```

Fig. 3.1 | New HTML5 form input types and attributes. (Part 2 of 5.)

```
48          <p>
49             <label>Month:
50                <input type = "month" /> (yyyy-mm)
51             </label>
52          </p>
53          <p>
54             <label>Number:
55                <input type = "number"
56                   min = "0"
57                   max = "7"
58                   step = "1"
59                   value = "4" />
60             </label> (Enter a number between 0 and 7)
61          </p>
62          <p>
63             <label>Range:
64                0 <input type = "range"
65                   min = "0"
66                   max = "20"
67                   value = "10" /> 20
68             </label>
69          </p>
```

Fig. 3.1 | New HTML5 form input types and attributes. (Part 3 of 5.)

```
70          <p>
71             <label>Search:
72                <input type = "search" placeholder = "search query" />
73             </label> (Enter your search query here.)
74          </p>
75          <p>
76             <label>Tel:
77                <input type = "tel" placeholder = "(###) ###-####"
78                   pattern = "\(\d{3}\) +\d{3}-\d{4}" required />
79                   (###) ###-####
80             </label>
81          </p>
82          <p>
83             <label>Time:
84                <input type = "time" /> (hh:mm:ss.ff)
85             </label>
86          </p>
87          <p>
88             <label>URL:
89                <input type = "url"
90                   placeholder = "http://www.domainname.com" />
91                   (http://www.domainname.com)
92             </label>
93          </p>
```

Fig. 3.1 | New HTML5 form input types and attributes. (Part 4 of 5.)

```
94            <p>
95               <label>Week:
96                  <input type = "week" />
97                     (yyyy-Wnn, such as 2012-W01)
98               </label>
99            </p>
100           <p>
101              <input type = "submit" value = "Submit" />
102              <input type = "reset" value = "Clear" />
103           </p>
104        </form>
105     </body>
106  </html>
```

Fig. 3.1 | New HTML5 form input types and attributes. (Part 5 of 5.)

3.1.1 input Type color

- The color input type enables the user to enter a color.
- At the time of this writing, most browsers render the `color input` type as a text field in which the user can enter a hexadecamal code or a color name.
- In the future, when you click a `color input`, browsers will likely display a *color picker* similar to the Microsoft Windows color dialog shown in Fig. 3.2.

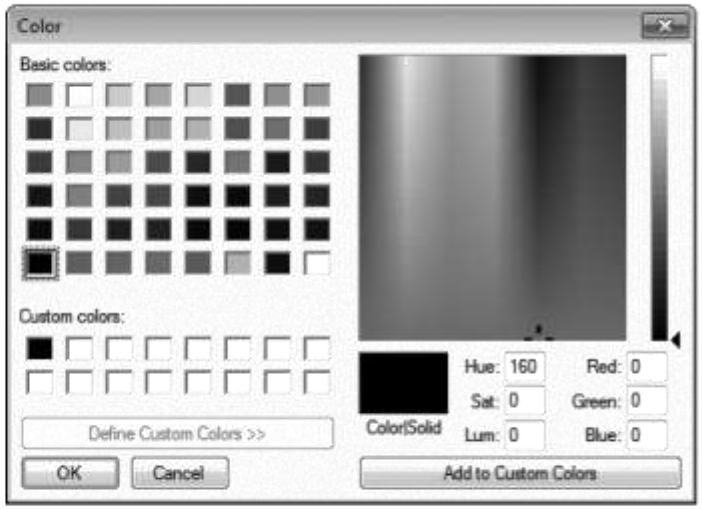

Fig. 3.2 | A dialog for choosing colors.

3.1.1 input Type color

autofocus Attribute

- The autofocus attribute—an optional attribute that can be used in only one `input` element on a form—automatically gives the focus to the `input` element, allowing the user to begin typing in that element immediately.

3.1.1 input Type color (cont.)

- Figure 3.3 shows autofocus on the color element—the first input element in our form—as rendered in Chrome. You do not need to include autofocus in your forms.

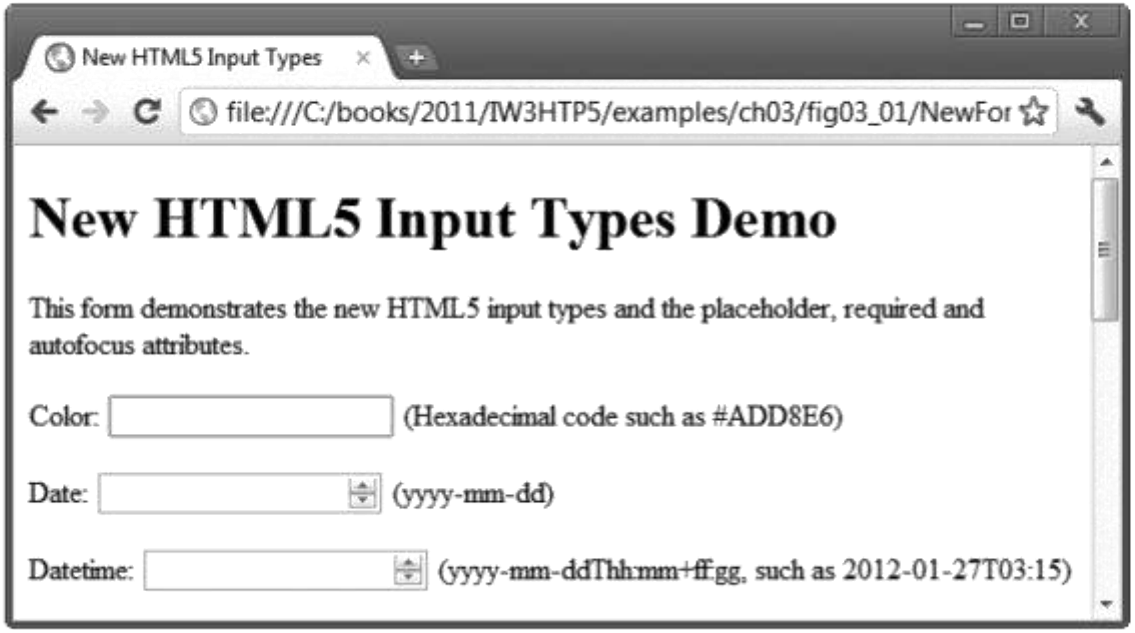

Fig. 3.3 | Autofocus in the `color input` element using Chrome.

3.1.1 input Type color (cont.)

Validation

- The new HTML 5 `input` types are *self validating* on the client side, eliminating the need to add complicated JavaScript code to your web pages to validate user input, reducing the amount of invalid data submitted and consequently reducing Internet traffic between the server and the client to correct invalid input.
- *The server should still validate all user input.*
- When a user enters data into a form then submits the form the browser immediately checks the self-validating elements to ensure that the data is correct (Fig. 3.4).

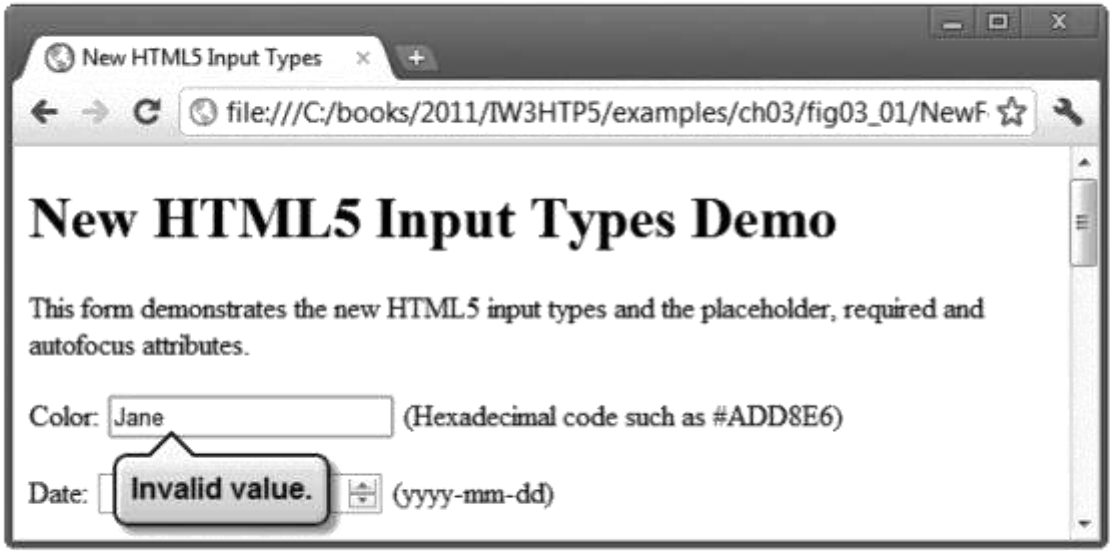

Fig. 3.4 | Validating a `color input` in Chrome.

3.1.1 input Type color (cont.)

- Figure 3.5 lists each of the new HTML5 input types and provides examples of the proper formats required for each type of data to be valid.

input type	Format
color	Hexadecimal code
date	yyyy-mm-dd
datetime	yyyy-mm-dd
datetime-local	yyyy-mm-ddThh:mm
month	yyyy-mm
number	Any numerical value
email	name@domain.com
url	http://www.domain-name.com
time	hh:mm
week	yyyy-Wnn

Fig. 3.5 | Self-validating `input` types.

3.1.1 input Type color (cont.)

- If you want to bypass validation, you can add the formnovalidate attribute to input type submit in line 101:

```
<input type = "submit" value = "Submit"
formnovalidate />
```

3.1.2 input Type date

- The date input type enables the user to enter a date in the form yyyy-mm-dd.
- Firefox and Internet Explorer display a text field in which a user can enter a date such as 2012-01-27.
- Chrome and Safari display a spinner control—a text field with an up-down arrow () on the right side—allowing the user to select a date by clicking the up or down arrow.
- The start date is the *current date*.
- Opera displays a calendar from which you can choose a date.
- In the future, when the user clicks a date input, browsers are likely to display a date control similar to the Microsoft Windows one shown in Fig. 3.6.

Fig. 3.6 | A date chooser control.

3.1.3 input Type datetime

- The datetime input type enables the user to enter a date (year, month, day), time (hour, minute, second, fraction of a second) and the time zone set to UTC (Coordinated Universal Time or Universal Time, Coordinated).
- Currently, most of the browsers render `datetime` as a text field; Chrome renders an up-down control and Opera renders a date and time control.

3.1.4 input Type datetime-local

- The datetime-local input type enables the user to enter the date and time in a *single* control.
- The data is entered as year, month, day, hour, minute, second and fraction of a second.
- Internet Explorer, Firefox and Safari all display a text field.
- Opera displays a date and time control.

3.1.5 input Type email

- The email input type enables the user to enter an e-mail address or a list of e-mail addresses separated by commas (if the `multiple` attribute is specified).
- Currently, all of the browsers display a text field.
- If the user enters an *invalid* e-mail address (i.e., the text entered is *not* in the proper format) and clicks the Submit button, a callout asking the user to enter an e-mail address is rendered pointing to the `input` element (Fig. 3.7).
- HTML5 does not check whether an e-mail address entered by the user actually exists—rather it just validates that the e-mail address is in the *proper format*.

Fig. 3.7 | Validating an e-mail address in Chrome.

3.1.5 input Type email (cont.)

placeholder Attribute

- The placeholder attribute allows you to place temporary text in a text field.
- Generally, placeholder text is *light gray* and provides an example of the text and/or text format the user should enter (Fig. 3.8).
- When the *focus* is placed in the text field (i.e., the cursor is in the text field), the placeholder text disappears—it's not "submitted" when the user clicks the Submit button (unless the user types the same text).

a) Text field with gray `placeholder` text

b) `placeholder` text disappears when the text field gets the focus

Fig. 3.8 | `placeholder` text disappears when the `input` element gets the focus.

3.1.5 input Type email (cont.)

- HTML5 supports placeholder text for only six input types—text, search, url, tel, email and password.

***required* Attribute**

- The required attribute forces the user to enter a value before submitting the form.
- You can add required to any of the input types.
- In this example, the user *must* enter an e-mail address and a telephone number to submit the form (Fig. 3.9).

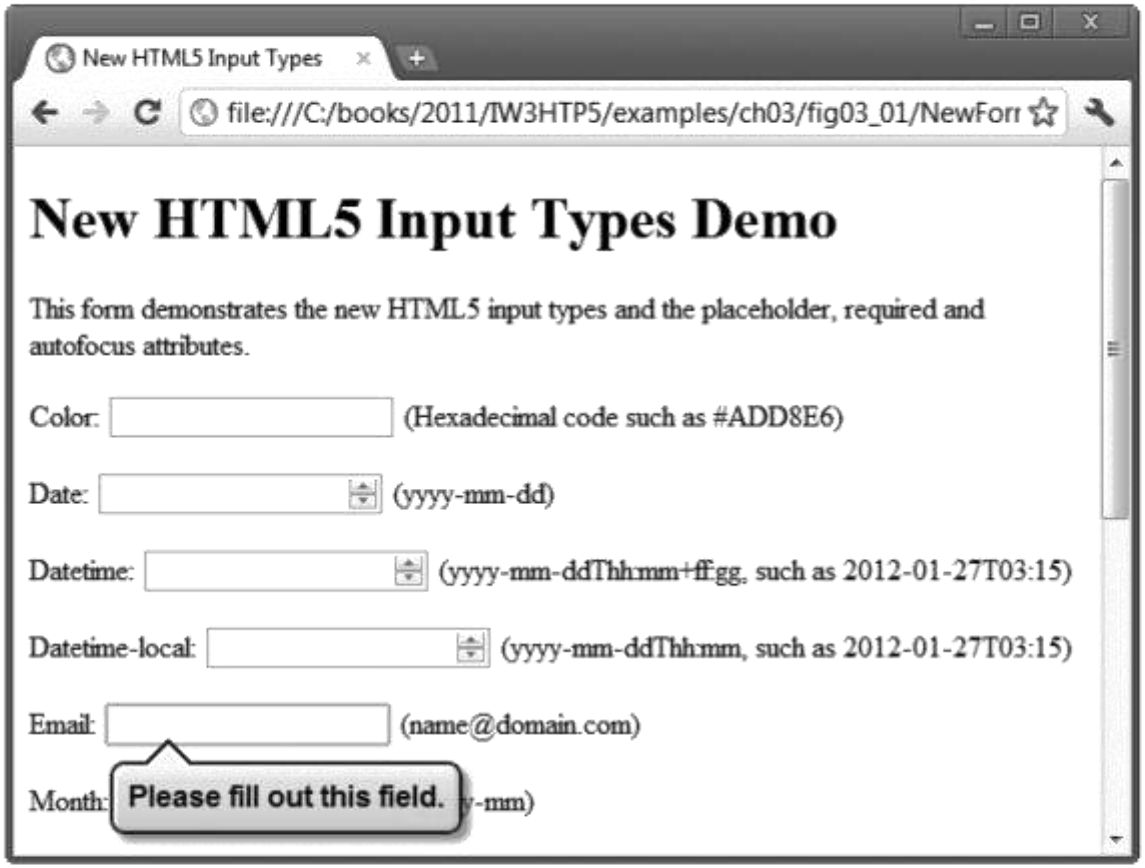

Fig. 3.9 | Demonstrating the `required` attribute in Chrome.

3.1.6 input Type month

- The month input type enables the user to enter a year and month in the format yyyy-mm, such as 2012-01.
- If the user enters the data in an improper format (e.g., January 2012) and submits the form, a callout stating that an invalid value was entered appears.

3.1.7 input Type number

- The number input type enables the user to enter a numerical value—mobile browsers typically display a numeric keypad for this `input` type.
- Internet Explorer, Firefox and Safari display a text field in which the user can enter a number. Chrome and Opera render a spinner control for adjusting the number.
- The `min` attribute sets the minimum valid number.
- The `max` attribute sets the maximum valid number.
- The `step` attribute determines the increment in which the numbers increase.
- The `value` attribute sets the initial value displayed in the form (Fig. 3.10).
- The spinner control includes only the valid numbers.
- If the user attempts to enter an invalid value by typing in the text field, a callout pointing to the number `input` element will instruct the user to enter a valid value.

Fig. 3.10 | `input` type `number` with a `value` attribute of 4 as rendered in Chrome.

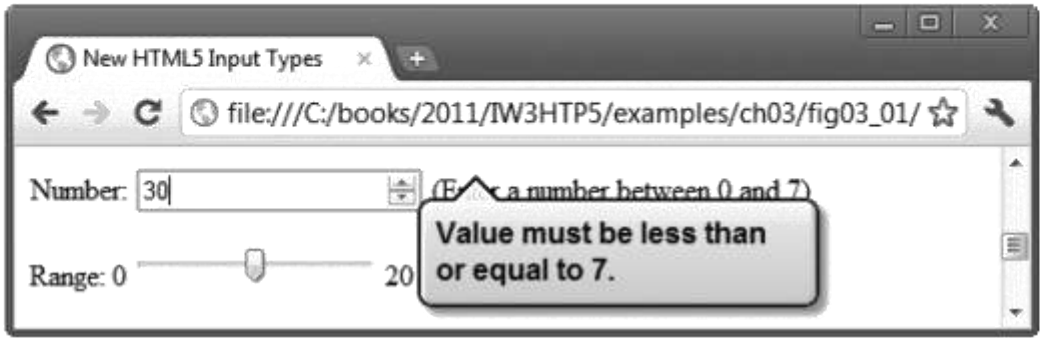

Fig. 3.11 | Chrome checking for a valid number.

3.1.8 input Type range

- The range input type appears as a *slider* control in Chrome, Safari and Opera (Fig. 3.12).
- You can set the minimum and maximum and specify a value.
- The range input type is *inherently self-validating* when it is rendered by the browser as a slider control, because *the user is unable to move the slider outside the bounds of the minimum or maximum value*.

Fig. 3.12 | range slider with a `value` attribute of 10 as rendered in Chrome.

3.1.9 input Type search

- The search input type provides a search field for entering a query.
- This `input` element is functionally equivalent to an `input` of type `text`.
- When the user begins to type in the search field, Chrome and Safari display an X that can be clicked to clear the field (Fig. 3.13).

Fig. 3.13 | Entering a search query in Chrome.

3.1.10 input Type tel

- The tel input type enables the user to enter a telephone number—mobile browsers typically display a keypad specific to entering phone numbers for this input type.
- At the time of this writing, the tel input type is rendered as a text field in all of the browsers.
- HTML5 does *not* self validate the tel input type.
- To ensure that the user enters a phone number in a proper format, we've added a pattern attribute that uses a *regular expression* to determine whether the number is in the format:
  - (555) 555-5555
- When the user enters a phone number in the wrong format, a callout appears requesting the proper format, pointing to the tel input element (Fig. 3.14).

Fig. 3.14 | Validating a phone number using the `pattern` attribute in the `tel input` type.

3.1.11 input Type time

- The time input type enables the user to enter an hour, minute, seconds and fraction of second (Fig. 3.15).
- The HTML5 specification indicates that a time must have two digits representing the hour, followed by a colon (:) and two digits representing the minute.
- Optionally, you can also include a colon followed by two digits representing the seconds and a period followed by one or more digits representing a fraction of a second (shown as ff in our sample text to the right of the time input element in Fig. 3.15.

Fig. 3.15 | `time` input as rendered in Chrome.

3.1.12 input Type url

- The url input type enables the user to enter a URL.
- The element is rendered as a text field, and the proper format is `http://www.deitel.com`.
- If the user enters an improperly formatted URL (e.g., `www.deitel.com` or `www.deitelcom`), the URL will *not* validate (Fig. 3.16).
- HTML5 does not check whether the URL entered is valid; rather it validates that the URL entered is in the proper format.

Fig. 3.16 | Validating a URL in Chrome.

3.1.13 input Type week

- The week input type enables the user to select a year and week number in the format yyyy-Wnn, where nn is 01–53—for example, 2012-W01 represents the first week of 2012. Internet Explorer, Firefox and Safari render a text field.
- Chrome renders an up-down control.
- Opera renders *week control* with a down arrow that, when clicked, brings up a calendar for the current month with the corresponding week numbers listed down the left side.

3.2 input and datalist Elements and autocomplete Attribute

- Figure 3.17 shows how to use the new autocomplete attribute and datalist element.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 3.17: autocomplete.html -->
4  <!-- New HTML5 form autocomplete attribute and datalist element. -->
5  <html>
6     <head>
7        <meta charset="utf-8">
8        <title>New HTML5 autocomplete Attribute and datalist Element</title>
9     </head>
10
11    <body>
12       <h1>Autocomplete and Datalist Demo</h1>
13       <p>This form demonstrates the new HTML5 autocomplete attribute
14          and the datalist element.
15       </p>
16
17       <!-- turn autocomplete on -->
18       <form method = "post" autocomplete = "on">
19          <p><label>First Name:
20             <input type = "text" id = "firstName"
21                placeholder = "First name" /> (First name)
22             </label></p>
```

Fig. 3.17 | New HTML5 form `autocomplete` attribute and `datalist` element. (Part 1 of 6.)

```
23          <p><label>Last Name:
24             <input type = "text" id = "lastName"
25                placeholder = "Last name" /> (Last name)
26             </label></p>
27          <p><label>Email:
28             <input type = "email" id = "email"
29                placeholder = "name@domain.com" /> (name@domain.com)
30             </label></p>
31          <p><label for = "txtList">Birth Month:
32             <input type = "text" id = "txtList"
33                placeholder = "Select a month" list = "months" />
34             <datalist id = "months">
35                <option value = "January">
36                <option value = "February">
37                <option value = "March">
38                <option value = "April">
39                <option value = "May">
40                <option value = "June">
41                <option value = "July">
42                <option value = "August">
43                <option value = "September">
44                <option value = "October">
45                <option value = "November">
```

Fig. 3.17 | New HTML5 form `autocomplete` attribute and `datalist` element. (Part 2 of 6.)

```
46                <option value = "December">
47             </datalist>
48          </label></p>
49          <p><input type = "submit" value = "Submit" />
50             <input type = "reset" value = "Clear" /></p>
51       </form>
52    </body>
53 </html>
```

Fig. 3.17 | New HTML5 form `autocomplete` attribute and `datalist` element. (Part 3 of 6.)

a) Form rendered in Firefox before the user interacts with it

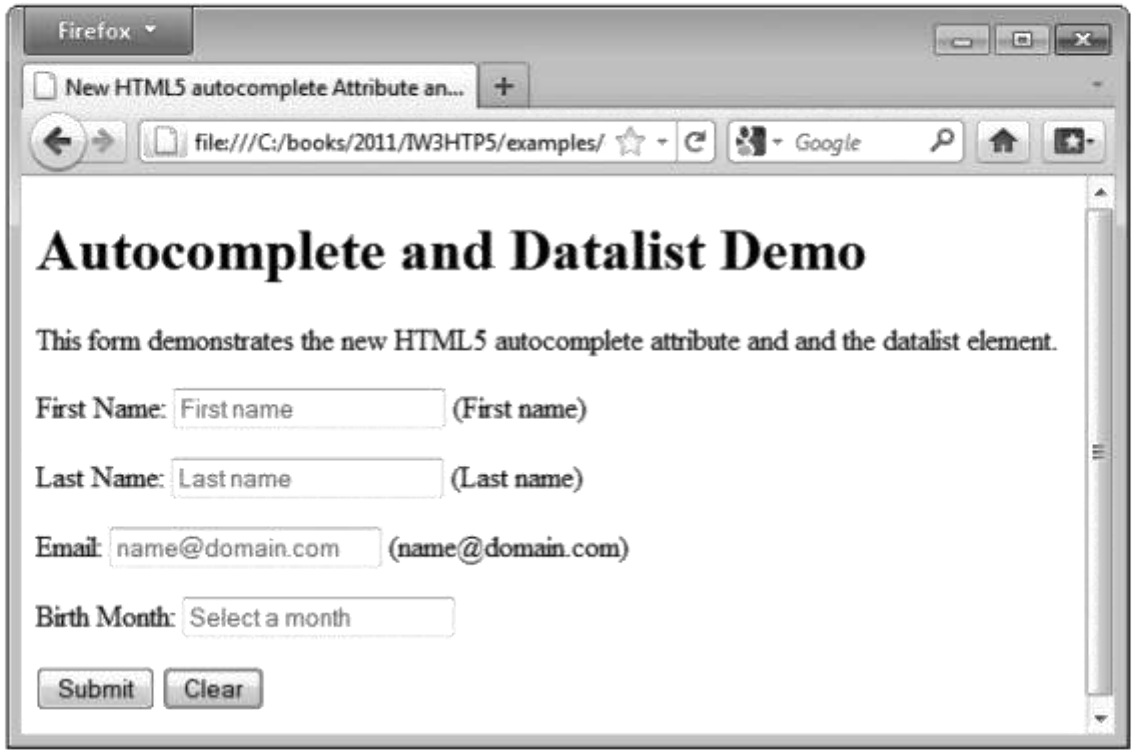

Fig. 3.17 | New HTML5 form `autocomplete` attribute and `datalist` element. (Part 4 of 6.)

b) `autocomplete` automatically fills in the data when the user returns to a form submitted previously and begins typing in the **First Name** `input` element; clicking `Jane` inserts that value in the `input`

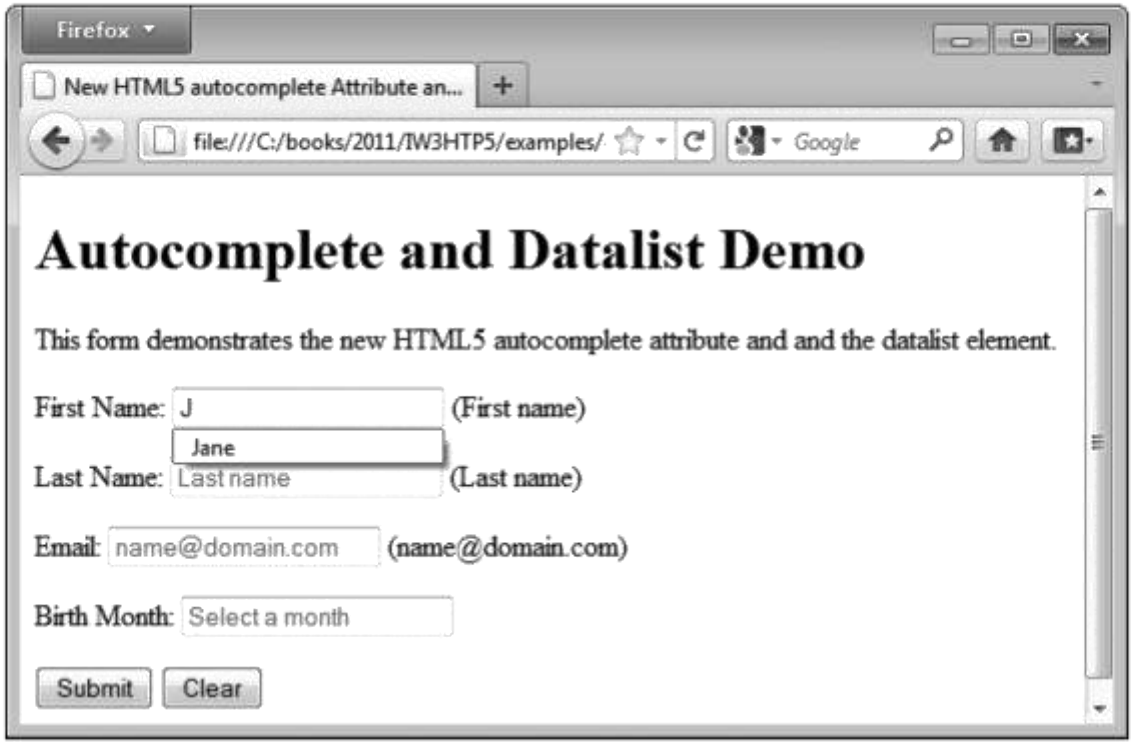

Fig. 3.17 | New HTML5 form `autocomplete` attribute and `datalist` element. (Part 5 of 6.)

c) `autocomplete` with a `datalist` showing the previously entered value (`June`) followed by all items that match what the user has typed so far; clicking an item in the `autocomplete` list inserts that value in the `input`

`datalist` values filtered by what's been typed so far

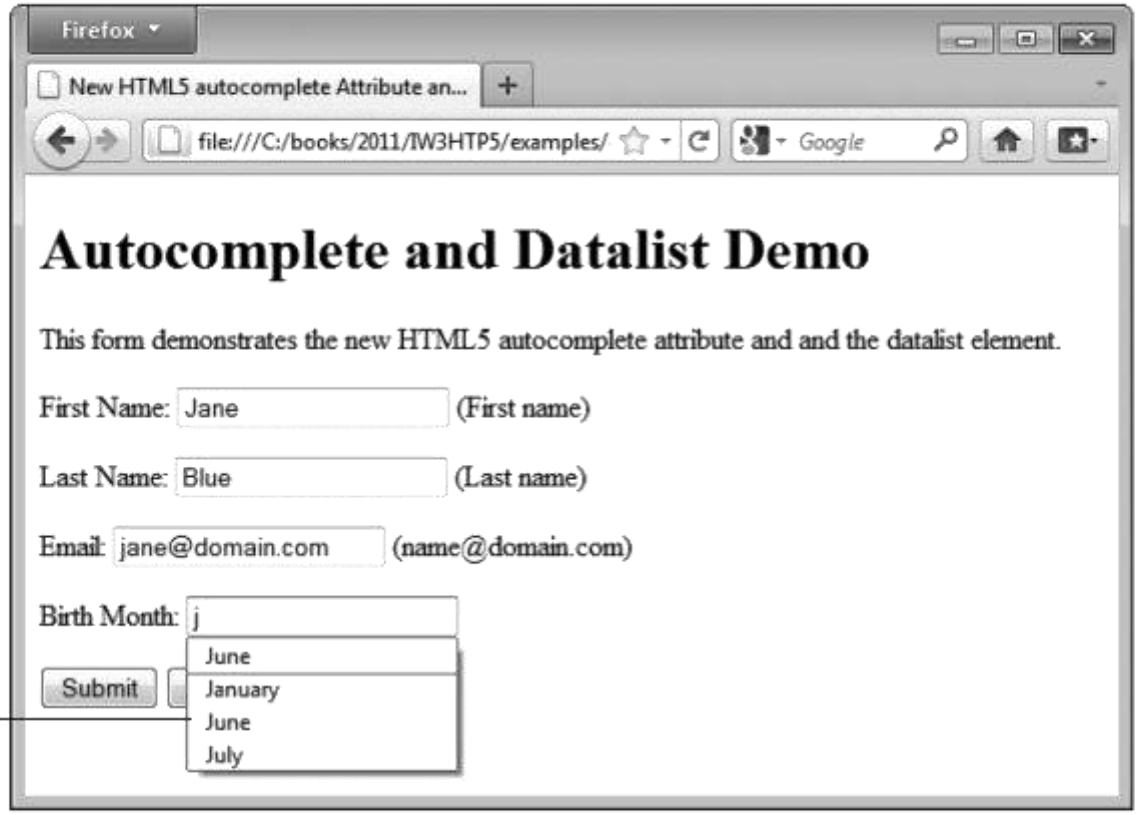

Fig. 3.17 | New HTML5 form `autocomplete` attribute and `datalist` element. (Part 6 of 6.)

3.2.1 input Element autocomplete Attribute

- The autocomplete attribute can be used on input types to automatically fill in the user's information based on previous input—such as name, address or e-mail.
- You can enable autocomplete for an entire form or just for specific elements.
- For example, an online order form might set automcomplete = "on" for the name and address inputs and set autocomplete = "off" for the credit card and password inputs for security purposes.

Error-Prevention Tip 3.1

The `autocomplete` attribute works only if you specify a `name` or `id` attribute for the `input` element.

3.2.2 datalist Element

- The datalist element provides input options for a `text input` element.
- At the time of this writing, `datalist` support varies by browser.
- In this example, we use a `datalist` element to obtain the user's birth month.
- Using Opera, when the user clicks in the text field, a drop-down list of the months of the year appears. If the user types `"M"` in the text field, the list on months is narrowed to March and May.
- When using Firefox, the drop-down list of months appears only after the user begins typing in the text field. If the user types `"M"`, all months containing the letter `"M"` or `"m"` appear in the drop-down list—March, May, September, November and December.