Chapter 3 Introduction to HTML5: Part 2

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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`

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```html
<!DOCTYPE html>

<!-- Fig. 3.1: newforminputtypes.html -->
<!-- New HTML5 form input types and attributes. -->
<html>
   <head>
      <meta charset="utf-8">
      <title>New HTML5 Input Types</title>
   </head>

   <body>
      <h1>New HTML5 Input Types Demo</h1>
      <p>This form demonstrates the new HTML5 input types
         and the placeholder, required and autofocus attributes.
      </p>

      <form method = "post" action = "http://www.deitel.com">
         <p>
            <label>Color:
               <input type = "color" autofocus />
                  (Hexadecimal code such as #ADD8E6)
            </label>
         </p>
```

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
            <p>
               <label>Date:
                  <input type = "date" />
                     (yyyy-mm-dd)
               </label>
            </p>
            <p>
               <label>Datetime:
                  <input type = "datetime" />
                     (yyyy-mm-ddThh:mm+ff:gg, such as 2012-01-27T03:15)
               </label>
            </p>
            <p>
               <label>Datetime-local:
                  <input type = "datetime-local" />
                     (yyyy-mm-ddThh:mm, such as 2012-01-27T03:15)
               </label>
            </p>
            <p>
               <label>Email:
                  <input type = "email" placeholder = "name@domain.com"
                     required /> (name@domain.com)
               </label>
            </p>
```

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
<p>
   <label>Month:
      <input type = "month" /> (yyyy-mm)
   </label>
</p>
<p>
   <label>Number:
      <input type = "number"
         min = "0"
         max = "7"
         step = "1"
         value = "4" />
   </label> (Enter a number between 0 and 7)
</p>
<p>
   <label>Range:
      0 <input type = "range"
         min = "0"
         max = "20"
         value = "10" /> 20
   </label>
</p>
```

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
<p>
   <label>Search:
      <input type = "search" placeholder = "search query" />
   </label> (Enter your search query here.)
</p>
<p>
   <label>Tel:
      <input type = "tel" placeholder = "(###) ###-####"
         pattern = "\(\d{3}\) +\d{3}-\d{4}" required />
         (###) ###-####
   </label>
</p>
<p>
   <label>Time:
      <input type = "time" /> (hh:mm:ss.ff)
   </label>
</p>
<p>
   <label>URL:
      <input type = "url"
         placeholder = "http://www.domainname.com" />
         (http://www.domainname.com)
   </label>
</p>
```

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
            <p>
               <label>Week:
                  <input type = "week" />
                     (yyyy-Wnn, such as 2012-W01)
               </label>
            </p>
            <p>
               <input type = "submit" value = "Submit" />
               <input type = "reset" value = "Clear" />
            </p>
         </form>
      </body>
   </html>
```

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Fig. 3.2 | A dialog for choosing colors.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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).

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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 />
```

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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*.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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).

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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).

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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*.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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).
-

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Fig. 3.13 | Entering a search query in Chrome.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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).
-

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.
-

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Fig. 3.16 | Validating a URL in Chrome.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.2 `input` and `datalist` Elements and `autocomplete` Attribute

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```html
<!DOCTYPE html>

<!-- Fig. 3.17: autocomplete.html -->
<!-- New HTML5 form autocomplete attribute and datalist element. -->
<html>
   <head>
      <meta charset="utf-8">
      <title>New HTML5 autocomplete Attribute and datalist Element</title>
   </head>

   <body>
      <h1>Autocomplete and Datalist Demo</h1>
      <p>This form demonstrates the new HTML5 autocomplete attribute
         and the datalist element.
      </p>

      <!-- turn autocomplete on -->
      <form method = "post" autocomplete = "on">
         <p><label>First Name:
            <input type = "text" id = "firstName"
               placeholder = "First name" /> (First name)
            </label></p>
```

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
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.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
                <option value = "December">
               </datalist>
            </label></p>
            <p><input type = "submit" value = "Submit" />
               <input type = "reset" value = "Clear" /></p>
         </form>
      </body>
   </html>
```

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Error-Prevention Tip 3.1

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

Copyright © Pearson, Inc. 2013. All Rights Reserved.

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.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3 Page-Structure Elements

- HTML5 introduces several new page-structure elements (Fig. 3.18) that meaningfully identify areas of the page as headers, footers, articles, navigation areas, asides, figures and more.
-

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```html
<!DOCTYPE html>

<!-- Fig. 3.18: sectionelements.html -->
<!-- New HTML5 section elements. -->
<html>
   <head>
      <meta charset="utf-8">
      <title>New HTML5 Section Elements</title>
   </head>

   <body>
      <header> <!-- header element creates a header for the page -->
         <img src = "deitellogo.png" alt = "Deitel logo" />
         <h1>Welcome to the Deitel Buzz Online<h1>

         <!-- time element inserts a date and/or time -->
         <time>2012-01-17</time>

      </header>

      <section id = "1"> <!-- Begin section 1 -->
         <nav> <!-- nav element groups navigation links  -->
            <h2> Recent Publications</h2>
```

Fig. 3.18 | New HTML5 section elements. (Part 1 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
            <ul>
               <li><a href = "http://www.deitel.com/books/iw3htp5">
                  Internet & World Wide Web How to Program, 5/e</a></li>
               <li><a href = "http://www.deitel.com/books/androidfp/">
                  Android for Programmers: An App-Driven Approach</a>
               </li>
               <li><a href = "http://www.deitel.com/books/iphonefp">
                  iPhone for Programmers: An App-Driven Approach</a></li>
               <li><a href = "http://www.deitel.com/books/jhtp9/">
                  Java How to Program, 9/e</a></li>
               <li><a href = "http://www.deitel.com/books/cpphtp8/">
                  C++ How to Program, 8/e</a></li>
               <li>
                  <a href = "http://www.deitel.com/books/vcsharp2010htp">
                     Visual C# 2010 How to Program, 4/e</a></li>
               <li><a href = "http://www.deitel.com/books/vb2010htp">
                  Visual Basic 2010 How to Program</a></li>
            </ul>
         </nav>
      </section>

      <section id = "2"> <!-- Begin section 2 -->
         <h2>How to Program Series Books</h2>
         <h3><em>Java How to Program, 9/e</em></h3>
```

Fig. 3.18 | New HTML5 section elements. (Part 2 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
48
49          <figure> <!-- figure element describes the image -->
50             <img src = "jhtp.jpg" alt = "Java How to Program, 9/e" />
51
52             <!-- figurecaption element inserts a figure caption -->
53             <figcaption><em>Java How to Program, 9/e</em>
54                cover.</figcaption>
55          </figure>
56
57          <!--article element represents content from another source -->
58          <article>
59             <header>
60                <h5>From
61                   <em>
62                      <a href = "http://www.deitel.com/books/jhtp9/">
63                      Java How to program, 9/e: </a>
64                   </em>
65                </h5>
66             </header>
67
```

Fig. 3.18 | New HTML5 section elements. (Part 3 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
            <p>Features include:
               <ul>
                  <li>Rich coverage of fundamentals, including
                     <!-- mark element highlights text -->
                     <mark>two chapters on control statements.</mark></li>
                  <li>Focus on <mark>real-world examples.</mark></li>
                  <li><mark>Making a Difference exercises set.</mark></li>
                  <li>Early introduction to classes, objects,
                     methods and strings.</li>
                  <li>Integrated exception handling.</li>
                  <li>Files, streams and object serialization.</li>
                  <li>Optional modular sections on language and
                     library features of the new Java SE 7.</li>
                  <li>Other topics include: Recursion, searching,
                     sorting, generic collections, generics, data
                     structures, applets, multimedia,
                     multithreading, databases/JDBC&trade;, web-app
                     development, web services and an optional
                     ATM Object-Oriented Design case study.</li>
               </ul>

```

Fig. 3.18 | New HTML5 section elements. (Part 4 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
<!-- summary element represents a summary for the -->
<!-- content of the details element -->
<details>
   <summary>Recent Edition Testimonials</summary>
   <ul>
      <li>"Updated to reflect the state of the
         art in Java technologies; its deep and
         crystal clear explanations make it
         indispensable. The social-consciousness
         [Making a Difference] exercises are
         something really new and refreshing."
         <strong>&mdash;Jos&eacute; Antonio
         Gonz&aacute;lez Seco, Parliament of
         Andalusia</strong></li>
      <li>"Gives new programmers the benefit of the
         wisdom derived from many years of software
         development experience."<strong>
         &mdash;Edward F. Gehringer, North Carolina
         State University</strong></li>
      <li>"Introduces good design practices and
         methodologies right from the beginning.
         An excellent starting point for developing
         high-quality robust Java applications."
         <strong>&mdash;Simon Ritter,
         Oracle Corporation</strong></li>
```

Fig. 3.18 | New HTML5 section elements. (Part 5 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
                    <li>"An easy-to-read conversational style.
                       Clear code examples propel readers to
                       become proficient in Java."
                       <strong>&mdash;Patty Kraft, San Diego State
                       University</strong></li>
                    <li>"A great textbook with a myriad of examples
                       from various application domains&mdash;
                       excellent for a typical CS1 or CS2 course."
                       <strong>&mdash;William E. Duncan, Louisiana
                       State University</strong></li>
                 </ul>
              </details>
           </p>
        </article>

        <!-- aside element represents content in a sidebar that's -->
        <!-- related to the content around the element -->
        <aside>
           The aside element is not formatted by the browsers.
        </aside>

```

Fig. 3.18 | New HTML5 section elements. (Part 6 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
         <h2>Deitel Developer Series Books</h2>
         <h3><em>Android for Programmers: An App-Driven Approach
            </em></h3>
            Click <a href = "http://www.deitel.com/books/androidfp/">
            here</a> for more information or to order this book.

         <h2>LiveLessons Videos</h2>
         <h3><em>C# 2010 Fundamentals LiveLessons</em></h3>
         Click <a href = "http://www.deitel.com/Books/LiveLessons/">
         here</a> for more information about our LiveLessons videos.
      </section>

      <section id = "3"> <!-- Begin section 3 -->
         <h2>Results from our Facebook Survey</h2>
         <p>If you were a nonprogrammer about to learn Java for the first
            time, would you prefer a course that taught Java in the
            context of Android app development? Here are the results from
            our survey:</p>

            <!-- meter element represents a scale within a range -->
            0 <meter min = "0"
            max = "54"
            value = "14"></meter> 54
```

Fig. 3.18 | New HTML5 section elements. (Part 7 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

```
         <p>Of the 54 responders, 14 (green) would prefer to
         learn Java in the context of Android app development.</p>
      </section>

      <!-- footer element represents a footer to a section or page, -->
      <!-- usually containing information such as author name, -->
      <!-- copyright, etc. -->
      <footer>
         <!-- wbr element indicates the appropriate place to break a -->
         <!-- word when the text wraps -->
         <h6>&copy; 1992-2012 by Deitel &amp; Associ<wbr>ates, Inc.
            All Rights Reserved.<h6>
         <!-- address element represents contact information for a -->
         <!-- document or the nearest body element or article -->
         <address>
            Contact us at <a href = "mailto:deitel@deitel.com">
            deitel@deitel.com</a>
         </address>
      </footer>
   </body>
</html>
```

Fig. 3.18 | New HTML5 section elements. (Part 8 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

a) Chrome browser showing the **header** element and a **nav** element that contains an unordered list of links

Fig. 3.18 | New HTML5 section elements. (Part 9 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

b) Chrome browser showing the beginning of a `section` containing a `figure` and a `figurecaption`

Fig. 3.18 | New HTML5 section elements. (Part 10 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

c) Chrome browser showing an `article` containing a `header`, some content and a collapsed `details` element, followed by an `aside` element

Fig. 3.18 | New HTML5 section elements. (Part 11 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

d) Chrome browser showing the end of the `section` that started in part (b)

Fig. 3.18 | New HTML5 section elements. (Part 12 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

e) Chrome browser showing the last `section` containing a `meter` element, followed by a `footer` element

Fig. 3.18 | New HTML5 section elements. (Part 13 of 13.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.1 header Element

- The header element creates a header for this page that contains both text and graphics.
- The header element can be used multiple times on a page and can include HTML headings (<h1> through <h6>), navigation, images and logos and more.

***time** Element*

- The time element, which does not need to be enclosed in a header, enables you to identify a date (as we do here), a time or both.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.2 nav Element

- The nav element groups navigation links.
- In this example, we used the heading Recent Publications and created a `ul` element with seven `li` elements that link to the corresponding web pages for each book.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.3 `figure` Element and `figcaption` Element

- The figure element describes a figure (such as an image, chart or table) in the document so that it could be moved to the side of the page or to another page.
- The figcaption element provides a caption for the image in the `figure` element.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.4 `article` Element

- The `article` element describes standalone content that could potentially be used or distributed elsewhere, such as a news article, forum post or blog entry.
- You can nest `article` elements. For example, you might have reader comments about a magazine nested as an `article` within the magazine `article`.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.5 summary Element and details Element

- The summary element displays a right-pointing arrow next to a summary or caption when the document is rendered in a browser (Fig. 3.19).
- When clicked, the arrow points downward and reveals the content in the details element.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Fig. 3.19 | Demonstrating the `summary` and `detail` elements. (Part 1 of 2.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Fig. 3.19 | Demonstrating the `summary` and `detail` elements. (Part 2 of 2.)

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.6 section Element

- The section element describes a section of a document, usually with a heading for each section—these elements can be nested.
- In this example, we broke the document into three sections—the first is Recent Publications.
- The section element may also be nested in an article.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.7 `aside` Element

- The aside element describes content that's related to the surrounding content (such as an `article`) but is somewhat separate from the flow of the text.
- For example, an `aside` in a news story might include some background history.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.8 `meter` Element

- The meter element renders a visual representation of a measure within a range (Fig. 3.20).
- In this example, we show the results of a recent web survey we did.
- The `min` attribute is `"0"` and a `max` attribute is `"54"` —indicating the total number of responses to our survey.
- The `value` attribute is `"14"`, representing the total number of people who responded “yes” to our survey question.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

Fig. 3.20 | Chrome rendering the `meter` element.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.9 `footer` Element

- The footer element describes a *footer*—content that usually appears at the bottom of the content or `section` element.
- In this example, we use the `footer` to describe the copyright notice and contact information.

Copyright © Pearson, Inc. 2013. All Rights Reserved.

3.3.10 Text-Level Semantics: mark Element and wbr Element

- The mark element highlights the text that's enclosed in the element.
- The wbr element indicates the appropriate place to break a word when the text wraps to multiple lines.
- You might use wbr to prevent a word from breaking in an awkward place.

Copyright © Pearson, Inc. 2013. All Rights Reserved.