

# CPIS 358: INTERNET APPLICATIONS<sup>1</sup> & WEB PROGRAMMING

CREATE

BEAUTIFUL

WEBSITES

GO>

## CHAPTER 2-PART 2: INTRODUCTION TO HTML5

# TEXTBOOK

2

- You can read Chapter 3 page 109-->136

# AGENDA

3

- Introduction
- HTML5 Form input Types
- Input and dataList Elements and autocomplete Attribute
- Page Structure Elements

# OBJECTIVES

4

- At the end of this chapter, the student should be able to:
  - ▣ Build a Form using the new HTML5 input types
  - ▣ Use self-validating input elements
  - ▣ Use autocomplete input
  - ▣ Use datalist
  - ▣ Use HTML5's new page-structure elements to delineate parts of page

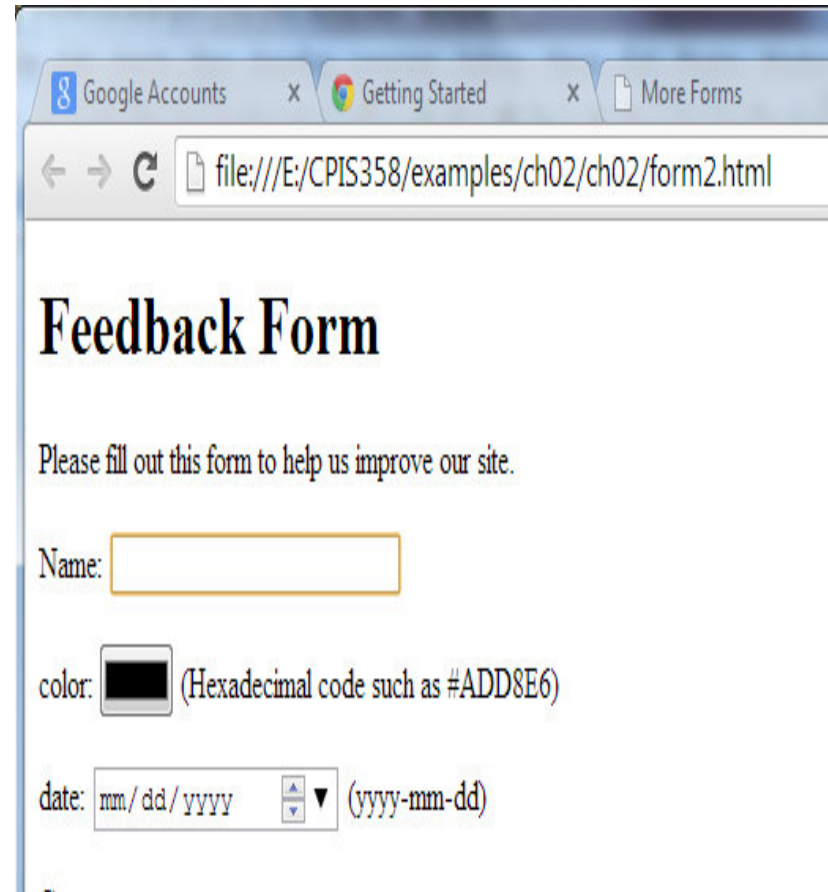
# New HTML5 Form input Types

- HTML5 defines new form input types

- These are not yet universally supported by all browsers such as Internet Explorer
- Input types:
  - Color, Date, Datetime, Datetime-local, Month, Week, Time, Email, Range, Number, Tel, url
- Autofocus attribute
- Self-validating input types

# The autofocus attribute of input element

- 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
  - **<input type="text" autofocus />**
  - This Figure shows autofocus on the Text element—the *first input element in our form*—as rendered in Chrome



The screenshot shows a web browser window with three tabs: 'Google Accounts', 'Getting Started', and 'More Forms'. The address bar displays the file path 'file:///E:/CPIS358/examples/ch02/ch02/form2.html'. The page content is titled 'Feedback Form' and includes the instruction 'Please fill out this form to help us improve our site.' The form contains three input fields: a text field for 'Name:', a color picker for 'color:', and a date selector for 'date:'. The 'Name:' text input field is highlighted with a thick orange border, indicating it has the autofocus attribute applied. The 'color:' field is a small square color picker, and the 'date:' field is a date selector with a dropdown arrow. The date field is currently set to 'mm/dd/yyyy' and has a hint '(yyyy-mm-dd)' next to it.

# Validation of Input Element...

- The new HTML 5 input types are self validating on the client side. This validation
  - ▣ eliminates the need to add complicated JavaScript code to your web pages to validate user input,
  - ▣ reduces the amount of invalid data submitted
  - ▣ and consequently reduces 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

# Validation of Input Element

- This figure lists each of the new HTML5 input types and provides examples of the proper formats required for each type of data to be valid
  - ▣ This is depend from language specified when you install your OS
- If you would to bypass validation, you can add the **formnovalidate** attribute to input type submit
  - ▣ `<input type = "submit" value = "Submit" formnovalidate />`

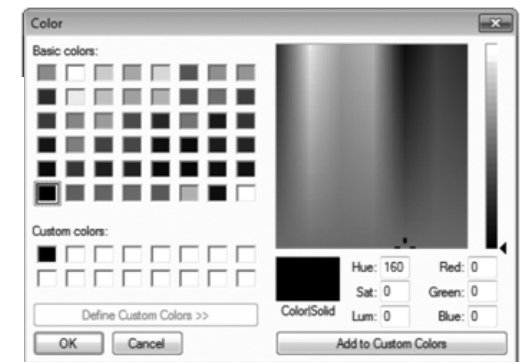
input type	Format
color	Hexadecimal code
date	yyyy-mm-dd
datetime	yyyy-mm-dd
datetime-local	yyyy-mm-ddThh:mm
month	yyyy-mm
number	Any numerical value
email	name@domain.com
url	http://www.domain-name.com
time	hh:mm
week	yyyy-Wnn



# 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 hexadecimal code or a color name
- With Google chrome, when you click a color input, browsers likely display a color picker similar to the Microsoft Windows color dialog shown in this figure

```
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>
```



| A dialog for choosing colors.

# Input Type date, datetime, datetime-local, month...

10

## New HTML5 Form Input Types

1 date

mm/dd/yyyy

Markup

```
<input type="date">
```

2 datetime

Markup

```
<input type="datetime">
```

3 datetime-local

mm/dd/yyyy --:--:--

Markup

```
<input type="datetime-local">
```

4 month

5 time

6 week

# 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`
  - ▣ 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*
  - ▣ Chrome and Opera display a calendar from which you can choose a date

# Input Type date

```
24 <p>
25   <label>Date:
26     <input type = "date" />
27       (yyyy-mm-dd)
28   </label>
29 </p>
```

date:  (yyyy-mm-dd)

Comm: September, 2013

Enter:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

E-mail:

Things you need:

# Input Type datetime

- The **datetime input type** enables the user to enter a **global** date and time string: 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

```
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>
```

# Input Type datetime-local

- The **datetime-local input type** enables the user to enter a date (year, month, day), time (hour, minute, second, fraction of a second)
- ▣ This input type is designed to take in a **local** date and time string

```
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>
```

# Input Type month

- The **month input type** enables the user to enter a year and month in the format understood by NLS language

48  
49  
50  
51  
52

- ▣ French : mm/yyyy such as 01/2012

- ▣ English:

- month, yyyy. Example: January 2012

- yyyy-mm, such as 2012-01

- ▣ If the data doesn't respect the format then a callout stating that an invalid value was entered appears, when the form is submitted

```
<p>  
  <label>Month:  
    <input type = "month" /> (yyyy-mm)  
  </label>  
</p>
```

The screenshot shows a web form with a date input field. The field is labeled "date:" and contains the text "November, 2013". To the right of the text are three small icons: a close button (X), a calendar icon, and a dropdown arrow. Below the input field, a calendar dropdown is open, showing a grid of months for the year 2013. The months are arranged in a 3x4 grid: Jan, Feb, Mar, Apr; May, Jun, Jul, Aug; Sep, Oct, Nov, Dec. The month "Nov" is highlighted with a blue background. To the left of the calendar, the year "2013" is displayed. Below the calendar, there are labels for "email:" and "Comm", and a text input field with the placeholder "Enter".

# Input Type time

- The **time input type** enables the user to enter an hour, minute, seconds and fraction of second
  - ▣ 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

```
<p>  
  <label>Time:  
    <input type = "time" /> (hh:mm:ss.ff)  
  </label>  
</p>
```

Time:

82  
83  
84  
85  
86  
--



# Input Type week

- The **week input type** enables the user to select a year and week number in the format **yyyy-Wnn** or **week nn, yyyy**
  - ▣ nn is the number of weeks
  - ▣ Example, 2012-W01 represents the first week of 2012
  - ▣ Internet Explorer, Firefox and Safari render a text field
  - ▣ Chrome, 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

94  
95  
96  
97  
98  
99

```
<p>  
  <label>Week:  
    <input type = "week" />  
    (yyyy-Wnn, such as 2012-W01)  
  </label>  
</p>
```

week: Week 39, 2013 x [up/down] ▼

email: September, 2013 ◀ ● ▶

Comm

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
36	1	2	3	4	5	6	7
37	8	9	10	11	12	13	14
38	15	16	17	18	19	20	21
39	22	23	24	25	26	27	28
40	29	30	1	2	3	4	5

Enter

# 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

email:  (name@domain.com)

- 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

email:  (name@domain.com)  
Comments: 

! Please enter an email address.

- 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*

# Input Type email...

## □ The email input type markup

```
2 <p>
3   <label>Email:
4     <input type = "email" placeholder = "name@domain.com"
5       required /> (name@domain.com)
6   </label>
7 </p>
```

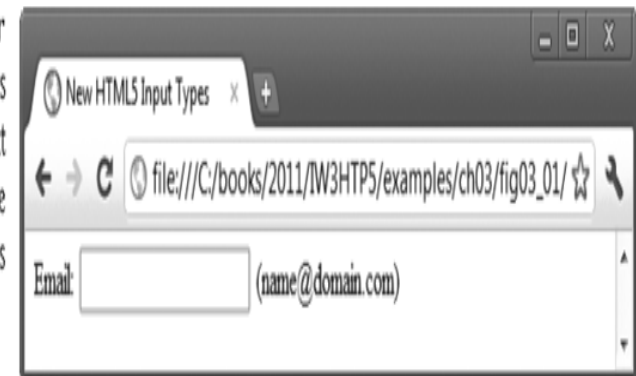
field

- Generally, placeholder text is *light gray* and provides an example of the text and/or text format the user should enter
- When the *focus* is placed in the text field (i.e., the cursor is in the text field), the placeholder text disappears

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



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



# Input Type email

- The **email input type** markup
  - ❑ `<input type = "email" placeholder="name@domain.com" required/>`
  - ❑ The **required attribute** forces the user to enter a value before submitting the form
- In this example, the user *must* enter an e-mail address and a telephone number to submit the form
- **You can add required to any input types**

New HTML5 Input Types Demo

This form demonstrates the new HTML5 input types and the placeholder, required and autofocus attributes.

Color:  (Hexadecimal code such as #ADD8E6)

Date:  (yyyy-mm-dd)

Datetime:  (yyyy-mm-ddThh:mm+ffgg, such as 2012-01-27T03:15)

Datetime-local:  (yyyy-mm-ddThh:mm, such as 2012-01-27T03:15)

Email:  (name@domain.com)

Month:  (y-mm)

Please fill out this field.

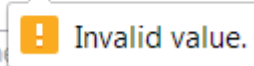
# Input Type number

21

- 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 that 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
  - **Attributes**
    - ▣ 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
- <input name = "number" type = "number" min="0" max="10" step="2" value="2"/>**

number:

number:

email:   Invalid value.

# Input Type range

22

- The **range input type** appears as a *slider* control in Chrome, Safari and Opera
  - ▣ You can set the minimum and maximum and specify a value
  - ▣ `<input name = "range" type = "range" min="0" max="20" value="10"/>`
- 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*

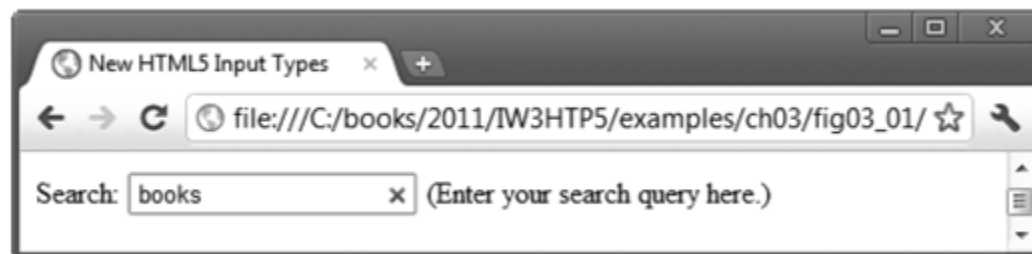


| range slider with a value attribute of 10 as rendered in

# Input Type search

23

- 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



# Input Type tel

24



- The **tel input type** enables the user to enter a telephone number
  - ❑ `<input name = "tel" type = "tel" placeholder="(####)###-####" pattern="\d{3}\d{3}-\d{4}"required/>`
  - ❑ At the time of this writing, the tel input type is rendered as a text field in all of the browsers
  - ❑ Mobile browsers typically display a keypad specific to entering phone numbers for this input type
  - ❑ 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



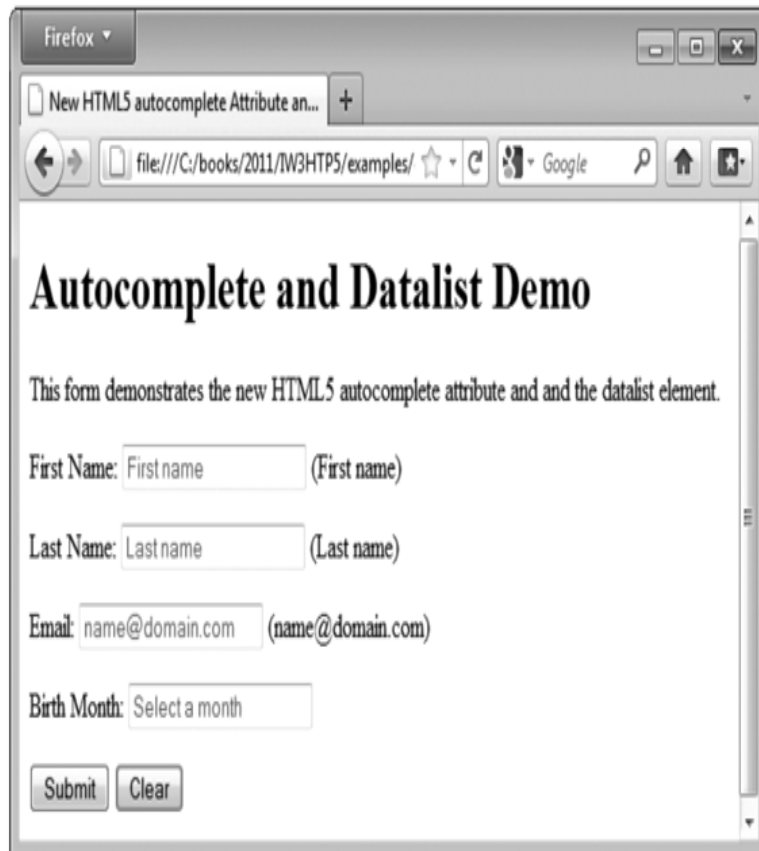
# Input Type URL

25

- 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.domainname.com`
  - ▣ If the user enters an improperly formatted URL (e.g., `www.deitel.com` or `www.deitelcom`), the URL will *not* validate
- HTML5 does not check whether the URL entered is valid; rather it validates that the URL entered is in the proper format



# Autocomplete Attribute



Firefox ▾

New HTML5 autocomplete Attribute an... +

file:///C:/books/2011/TW3HTPS/examples/ ☆ ↻ Google 🔍 🏠 ➦

## Autocomplete and Datalist Demo

This form demonstrates the new HTML5 autocomplete attribute and the datalist element.

First Name:  (First name)

Last Name:  (Last name)

Email:  (name@domain.com)

Birth Month:  Select a month



Firefox ▾

New HTML5 autocomplete Attribute an... +

file:///C:/books/2011/TW3HTPS/examples/ ☆ ↻ Google 🔍 🏠 ➦

## Autocomplete and Datalist Demo

This form demonstrates the new HTML5 autocomplete attribute and the datalist element.

First Name:  J (First name)

Last Name:  Jane (Last name)

Email:  (name@domain.com)

Birth Month:  Select a month

# Autocomplete Attribute

27

- 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
- The autocomplete attribute can be enabled
  - ▣ For an entire form
  - ▣ For specific elements
- For example, an online order form might set `autocomplete = "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.

```
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.)

# Datalist Element...

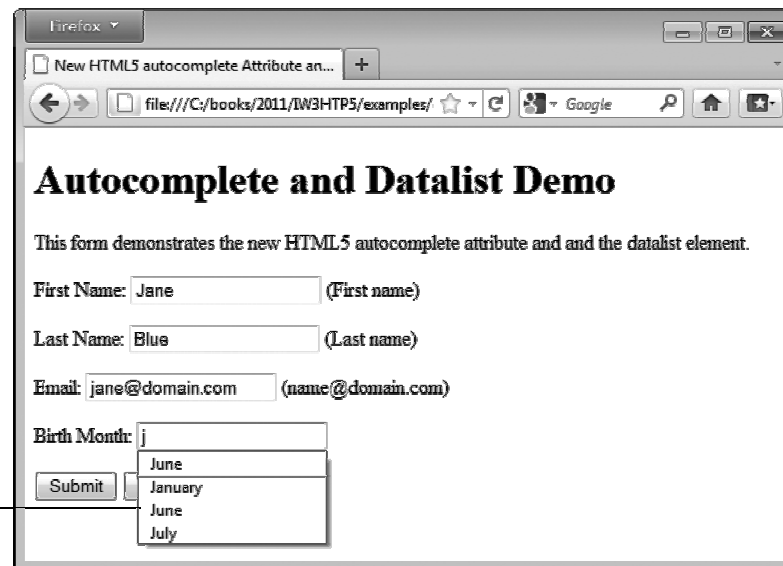
29

- 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.

# Datalist Element...

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



The screenshot shows a Firefox browser window with a new tab titled "New HTML5 autocomplete Attribute an...". The address bar shows a file path: "file:///C:/books/2011/TW3HTP5/examples/". The page content is titled "Autocomplete and Datalist Demo" and includes a description: "This form demonstrates the new HTML5 autocomplete attribute and the datalist element." The form contains four input fields: "First Name: Jane (First name)", "Last Name: Blue (Last name)", "Email: jane@domain.com (name@domain.com)", and "Birth Month: j". The "Birth Month" field is open, showing a list of months: "June", "January", "June", and "July". A "Submit" button is located to the left of the "Birth Month" field.

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

# Datalist Element

31

```
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">
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>
```

# Page-Structure Elements

32

- HTML5 introduces several new page-structure elements that meaningfully identify areas of the page as
  - ▣ Header Element
  - ▣ Nav Element
  - ▣ Figure Element and figcaption Element
  - ▣ Article Element
  - ▣ Summary Element and details Element
  - ▣ Section Element
  - ▣ Aside Element
  - ▣ Meter Element
  - ▣ Footer element
  - ▣ Mark and wbr Element

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





# 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
    - logos
    - Time element and more
      - 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

# Header Element

34

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 3.18: sectionelements.html -->
4  <!-- New HTML5 section elements. -->
5  <html>
6    <head>
7      <meta charset="utf-8">
8      <title>New HTML5 Section Elements</title>
9    </head>
10
11   <body>
12     <header> <!-- header element creates a header for the page -->
13       <img src = "deitellogo.png" alt = "Deitel logo" />
14       <h1>Welcome to the Deitel Buzz Online</h1>
15
16       <!-- time element inserts a date and/or time -->
17       <time>2012-01-17</time>
18
19     </header>
```

# 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

# 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

# nav Element

37

```
21 <section id = "1"> <!-- Begin section 1 -->
22 <nav> <!-- nav element groups navigation links -->
23 <h2> Recent Publications</h2>
24 <ul>
25 <li><a href = "http://www.deitel.com/books/iw3http5">
26 Internet & World Wide Web How to Program, 5/e</a></li>
27 <li><a href = "http://www.deitel.com/books/androidfp">
28 Android for Programmers: An App-Driven Approach</a>
29 </li>
30 <li><a href = "http://www.deitel.com/books/iphonefp">
31 iPhone for Programmers: An App-Driven Approach</a></li>
32 <li><a href = "http://www.deitel.com/books/jhttp9/">
33 Java How to Program, 9/e</a></li>
34 <li><a href = "http://www.deitel.com/books/cpphttp8/">
35 C++ How to Program, 8/e</a></li>
36 <li>
37 <a href = "http://www.deitel.com/books/vcsharp2010http">
38 Visual C# 2010 How to Program, 4/e</a></li>
39 <li><a href = "http://www.deitel.com/books/vb2010http">
40 Visual Basic 2010 How to Program</a></li>
41 </ul>
42 </nav>
43 </section>
```

# 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

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



# figure Element and figcaption Element

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

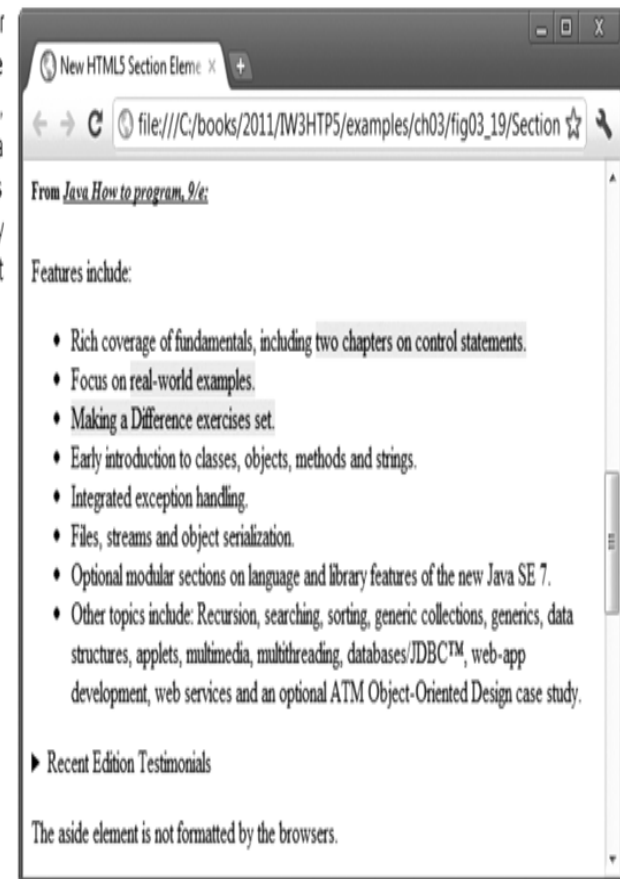
b) Chrome browser  
showing the  
beginning of a  
section containing  
a figure and a  
figurecaption



# article Element...

- The **article** element describes standalone content that could potentially be used or distributed elsewhere, such as a new 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**

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





# article Element

```
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
68          <p>Features include:
69              <ul>
70                  <li>Rich coverage of fundamentals, including
71                      <!-- mark element highlights text -->
72                      <mark>two chapters on control statements.</mark></li>
73                  <li>Focus on <mark>real-world examples.</mark></li>
74                  <li><mark>Making a Difference exercises set.</mark></li>
75                  <li>Early introduction to classes, objects,
76                      methods and strings.</li>
77                  <li>Integrated exception handling.</li>
78                  <li>Files, streams and object serialization.</li>
79                  <li>Optional modular sections on language and
80                      library features of the new Java SE 7.</li>
81                  <li>Other topics include: Recursion, searching,
82                      sorting, generic collections, generics, data
83                      structures, applets, multimedia,
84                      multithreading, databases/JDBC&trade;, web-app
85                      development, web services and an optional
86                      ATM Object-Oriented Design case study.</li>
87              </ul>
88
```

# aside Element

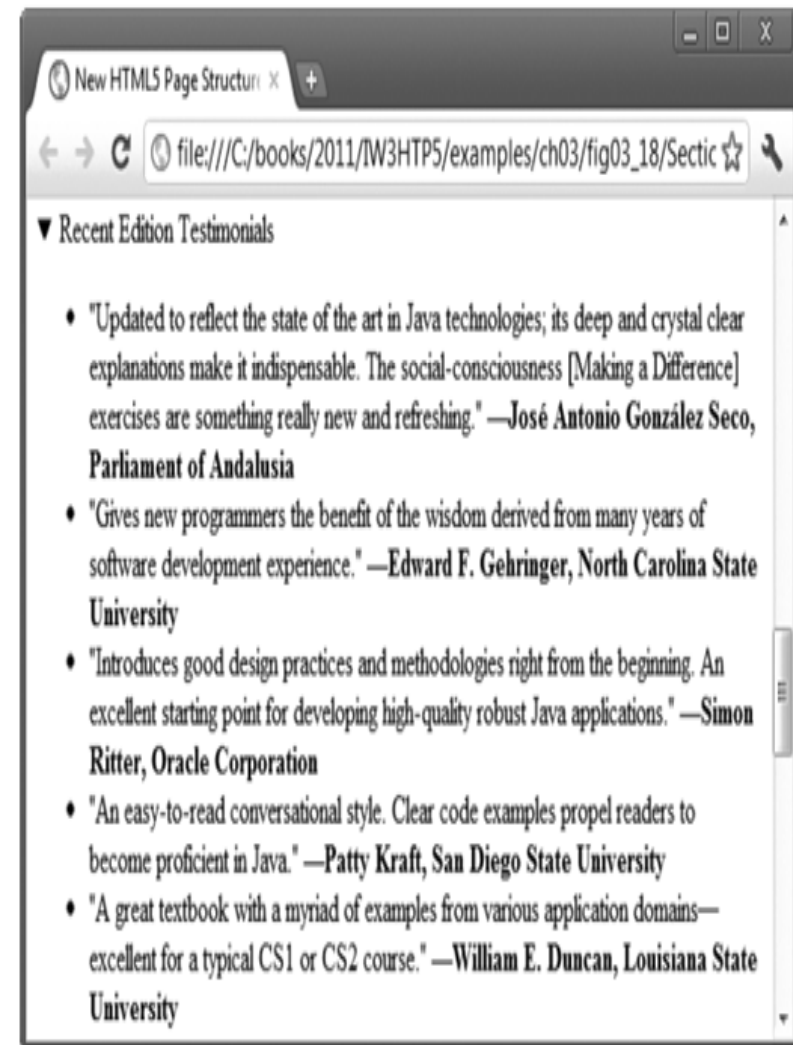
- The **aside element** describes content that's related to the surrounding content (such as an article) but is somewhat separate from the flow of the text

```
129      <!-- aside element represents content in a sidebar that's -->
130      <!-- related to the content around the element -->
131      <aside>
132          The aside element is not formatted by the browsers.
133      </aside>
134
```

- For example, an **aside** in a new story might include some background history

# 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
- When clicked, the arrow points downward and reveals the content in the **details element**



```

89      <!-- summary element represents a summary for the -->
90      <!-- content of the details element -->
91      <details>
92          <summary>Recent Edition Testimonials</summary>
93          <ul>
94              <li>"Updated to reflect the state of the
95                  art in Java technologies; its deep and
96                  crystal clear explanations make it
97                  indispensable. The social-consciousness
98                  [Making a Difference] exercises are
99                  something really new and refreshing."
100              <strong>&mdash;Jos&eacute; Antonio
101                  Gonz&aacute;lez Seco, Parliament of
102                  Andalusia</strong></li>
103              <li>"Gives new programmers the benefit of the
104                  wisdom derived from many years of software
105                  development experience."<strong>
106                  &mdash;Edward F. Gehringer, North Carolina
107                  State University</strong></li>
108              <li>"Introduces good design practices and
109                  methodologies right from the beginning.
110                  An excellent starting point for developing
111                  high-quality robust Java applications."
112              <strong>&mdash;Simon Ritter,
113                  Oracle Corporation</strong></li>
114
115              <li>"An easy-to-read conversational style.
116                  Clear code examples propel readers to
117                  become proficient in Java."
118              <strong>&mdash;Patty Kraft, San Diego State
119                  University</strong></li>
120              <li>"A great textbook with a myriad of examples
121                  from various application domains&mdash;
122                  excellent for a typical CS1 or CS2 course."
123              <strong>&mdash;William E. Duncan, Louisiana
124                  State University</strong></li>
125          </ul>
126      </details>

```

# meter Element...

- The **meter element** renders a visual representation of a measure within a range
- 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



# meter Element

46

```
147 <section id = "3"> <!-- Begin section 3 -->
148 <h2>Results from our Facebook Survey</h2>
149 <p>If you were a nonprogrammer about to learn Java for the first
150 time, would you prefer a course that taught Java in the
151 context of Android app development? Here are the results from
152 our survey:</p>
153
154 <!-- meter element represents a scale within a range -->
155 0 <meter min = "0"
156 max = "54"
157 value = "14"></meter> 54
158
159 <p>Of the 54 responders, 14 (green) would prefer to
160 learn Java in the context of Android app development.</p>
</section>
```

# 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

© 1992-2012 by Deitel & Associates, Inc. All Rights Reserved.

Contact us at [deitel@deitel.com](mailto:deitel@deitel.com)

# footer Element

```
161
162      <!-- footer element represents a footer to a section or page, -->
163      <!-- usually containing information such as author name, -->
164      <!-- copyright, etc. -->
165      <footer>
166          <!-- wbr element indicates the appropriate place to break a -->
167          <!-- word when the text wraps -->
168          <h6>&copy; 1992-2012 by Deitel & Associates, Inc.
169              All Rights Reserved.</h6>
170          <!-- address element represents contact information for a -->
171          <!-- document or the nearest body element or article -->
172          <address>
173              Contact us at <a href = "mailto:deitel@deitel.com">
174                  deitel@deitel.com</a>
175          </address>
176      </footer>
177  </body>
178 </html>
```



## 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

# References

50

- TextBook and Pearson slides
- Wikipedia

