



HTML: Hypertext Markup Language

CCSW 321 (Web Development)

What will be covered

- HTML Tables
- HTML forms and elements.
- Text autocomplete.
- HTML data validation layer.
- HTML5 Semantic tags.
- Website accessibility.

HTML Tables

- **Tables** are a fundamental way to **organize and present** data in a **structured** format on web pages.
- Tables organize data into **rows** and **columns**.
 - HTML uses the `<table>` element to create tables.
 - Tables are built using **rows** `<tr>` and **cells** `<td>` within those rows.
 - **Headers** are defined using the `<th>` element.
- The **caption** element specifies a table's title.

HTML Tables

A **table** can be **split** into three distinct sections:

- **Table Head (thead element)**
 - Table titles
 - Column headers
- **Table Body (tbody element)**
 - Primary table data
- **Table Foot (tfoot element)**
 - Calculation results
 - Footnotes
 - Above body section in the code, but displays at the bottom in the page

HTML Tables

- **Table Attributes**

- **border**: Specifies the border width.
- **width**: Defines the width of the table.
- **cellspacing**: Defines space between cells.
- **cellpadding**: Defines space between cell content and cell borders.

- **Spanning Cells:**

- **colspan**: Merges cells horizontally.
- **rowspan**: Merges cells vertically.

HTML Tables

- **Table Accessibility:**
 - Proper use of **header cells** (`<th>`) improves **accessibility**.
 - The **summary** attribute summarizes the table's contents and is used by speech devices to make the table more **accessible** to users with **visual impairments**.
 - Include a general **description** of a table's information in the table element's summary attribute—one of the many HTML5 features that make web pages **more accessible to users with disabilities**.
 - Speech devices use this attribute to make the table more accessible to users with visual impairments.

HTML Tables: Example

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.12: table1.html -->
4  <!-- Creating a basic table. -->
5  <html>
6    <head>
7      <meta charset = "utf-8">
8      <title>A simple HTML5 table</title>
9    </head>
10
11   <body>
12     <!-- the <table> tag opens a table -->
13     <table border = "1">
14
15       <!-- the <caption> tag summarizes the table's -->
16       <!-- contents (this helps visually impaired people) -->
17       <caption><strong>Table of Fruits (1st column) and
18         Their Prices (2nd column)</strong></caption>
19
20
21       <!-- the <thead> section appears first in the table -->
22       <!-- it formats the table header area -->
23       <thead>
24         <tr> <!-- <tr> inserts a table row -->
25           <th>Fruit</th> <!-- insert a heading cell -->
26           <th>Price</th>
27         </tr>
28       </thead>
29
30       <!-- the <tfoot> section appears last in the table -->
31       <!-- it formats the table footer -->
32       <tfoot>
33         <tr>
34           <th>Total</th>
35           <th>$3.75</th>
36         </tr>
37
38       </tfoot>
39
40       <!-- all table content is enclosed -->
41       <!-- within the <tbody> -->
42       <tbody>
43         <tr> by default
44           <td>Apple</td> <!-- insert a data cell -->
45             <td>$0.25</td>
46           </tr>
47
48         <tr>
49           <td>Orange</td>
50             <td>$0.50</td>
51           </tr>
52
53         <tr>
54           <td>Banana</td>
55             <td>$1.00</td>
56           </tr>
57
58         <tr>
59           <td>Pineapple</td>
60             <td>$2.00</td>
61           </tr>
62
63       </tbody>
64     </table>
65   </body>
66 </html>
```

HTML Tables: Example

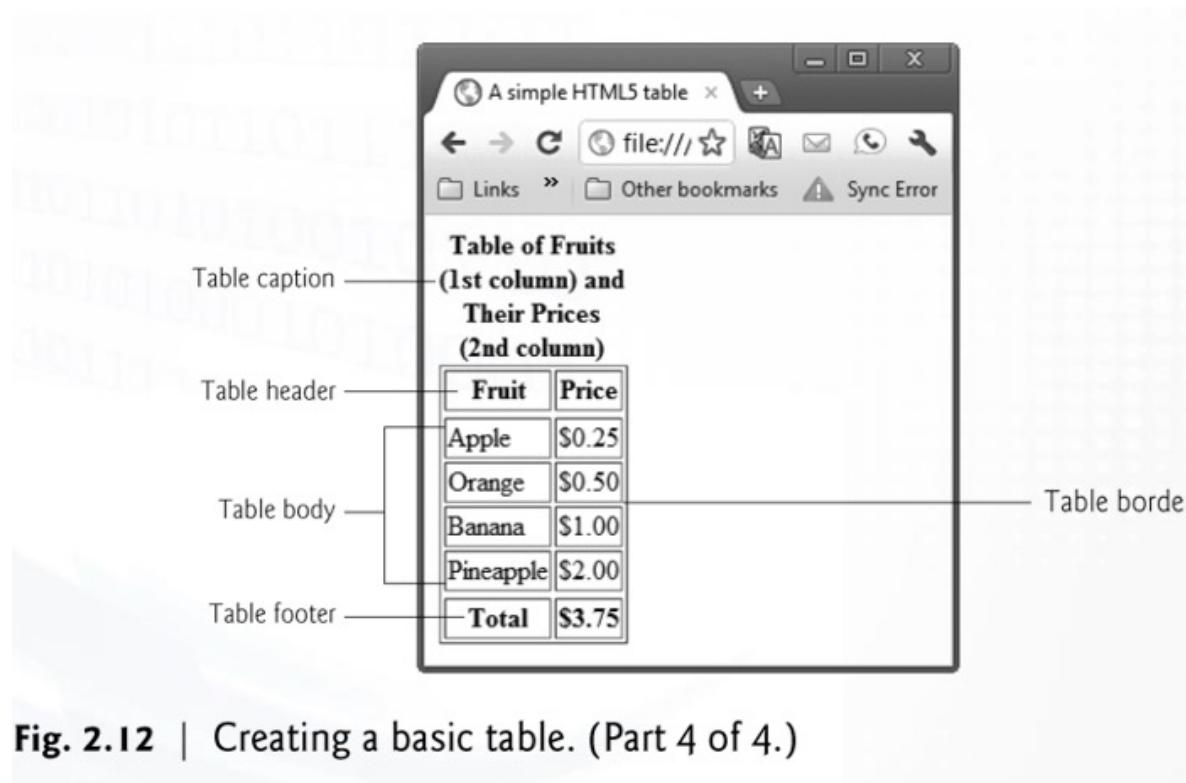


Fig. 2.12 | Creating a basic table. (Part 4 of 4.)

HTML Tables: Example

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.13: table2.html --&gt;
4  &lt;!-- Complex HTML5 table. --&gt;
5  &lt;html&gt;
6      &lt;head&gt;
7          &lt;meta charset = "utf-8"&gt;
8          &lt;title&gt;Tables&lt;/title&gt;
9      &lt;/head&gt;
10
11     &lt;body&gt;
12         &lt;h1&gt;Table Example: Spanning Rows and Columns&lt;/h1&gt;
13
14         &lt;table border = "1"&gt;
15             &lt;caption&gt;A more complex sample table&lt;/caption&gt;
16
17             &lt;thead&gt;
18                 &lt;!-- rowspans and colspans merge the specified --&gt;
19                 &lt;!-- number of cells vertically or horizontally --&gt;
20                 &lt;tr&gt;
21                     &lt;!-- merge two rows --&gt;
22                     &lt;th rowspan = "2"&gt;
23                         &lt;img src = "camel.png" width = "205"
24                         height = "167" alt = "Picture of a camel"&gt;
25                     &lt;/th&gt;
26
27                     &lt;!-- merge four columns --&gt;
28                     &lt;th colspan = "4"&gt;
29                         &lt;strong&gt;Camelid comparison&lt;/strong&gt;&lt;br&gt;
30                         Approximate as of 10/2011
31                     &lt;/th&gt;
32
33             &lt;/tr&gt;
34             &lt;tr&gt;
35                 &lt;th&gt;# of humps&lt;/th&gt;
36                 &lt;th&gt;Indigenous region&lt;/th&gt;
37                 &lt;th&gt;Spits?&lt;/th&gt;
38                 &lt;th&gt;Produces wool?&lt;/th&gt;
39             &lt;/tr&gt;
40             &lt;/thead&gt;
41
42             &lt;tbody&gt;
43                 &lt;tr&gt;
44                     &lt;th&gt;Camels (bactrian)&lt;/th&gt;
45                     &lt;td&gt;2&lt;/td&gt;
46                     &lt;td&gt;Africa/Asia&lt;/td&gt;
47                     &lt;td&gt;Yes&lt;/td&gt;
48                     &lt;td&gt;Yes&lt;/td&gt;
49             &lt;/tr&gt;
50             &lt;tr&gt;
51                 &lt;th&gt;Llamas&lt;/th&gt;
52                 &lt;td&gt;1&lt;/td&gt;
53                 &lt;td&gt;Andes Mountains&lt;/td&gt;
54                 &lt;td&gt;Yes&lt;/td&gt;
55                 &lt;td&gt;Yes&lt;/td&gt;
56             &lt;/tr&gt;
57         &lt;/tbody&gt;
58     &lt;/table&gt;
59
60     &lt;/body&gt;
61 &lt;/html&gt;</pre>
```

HTML Tables: Example



Tables

file:///C:/books/2011/IW3HTP5/examples/ch02/table2.html

Table Example: Spanning Rows and Columns

A more complex sample table

Camelid comparison Approximate as of 6/2011				
	# of humps	Indigenous region	Spits?	Produces wool?
Camels (bactrian)	2	Africa/Asia	Yes	Yes
Llamas	1	Andes Mountains	Yes	Yes



Fig. 2.13 | Complex HTML5 table. (Part 4 of 4.)

HTML Forms

- Forms enable us to collect information from users. For example, to take orders, fill surveys, or process user registration and so on.
- Visitor will fill in a form then click a button to submit.
- Submitted forms can be processed on the client side (e.g., JavaScript functions), or server-side (e.g., PHP code).
- **<form> element** is not visible, it represents an area that can contains form elements (other tags).

T.F

<form> Attributes

- **Attribute *method*** specifies how the form's data is sent to the web server. Using **method = "POST"** appends form data to the browser request, which contains the protocol (HTTP) and the requested resource's URL.
URL یعنی لینک
- The other possible value, **method = "GET"**, appends the form data directly to the end of the URL of the script, where it's visible in the browser's Address field. **GET** is the default method
- The ***action attribute*** of the form element specifies where the information should be sent when the user submit the form.

HTML Forms Elements

- Forms can contain **visual** and **nonvisual** components.
- **Visual components** include clickable buttons and other graphical user interface components with which users interact.
- **Nonvisual components**, called hidden inputs, store any data that you specify, such as e-mail addresses and HTML5 document file names that act as links.
- Every form related element should have a **name** attribute that helps the **receiving** program **identify** the information.

HTML Forms Elements

- The HTML <form> element can contain one or more form elements. For example:
- <**label**>: The element defines a label for a specific form element. تُنَامِيلُ الْفَرْجَ بِي
يَعْلَمُ الْجَوْزَ
 - The element is **useful** for **screen-reader** users, because the screen-reader will read out loud the label when the user focus on the input element.
- <**select**>: defines a drop-down list.
- <**textarea**>: defines a multi-line input field.
 - The number of rows is specified with the rows attribute, and the number of columns (i.e., characters per line) with the cols attribute

HTML Forms: Elements

- **<input>**: One of the most used form element
 - The element can be displayed in several ways, depending on the **type** attribute.

HTML Input Types

Here are the different input types you can use in HTML:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`



Form Input types

Type= text

Type= tel

Type= password

Type= checkbox



خيار واحد

Type= radio



Type= date

Type= color

Type= file

Type= range

Type= submit

Submit

HTML Forms Elements

- Type **text** input element enables the collection of textual user input.
- Type **checkbox** input element enables select options.
 - If selected, a check mark appears in the checkbox.
 - **checked** attribute make the option pre-checked.
 - checkboxes of same group have same name.
- Type **radio** are similar to checkboxes, except only one radio button in a group is selected at any time.
 - All radio buttons of a group have the same name attribute.

HTML Forms Elements



Common Programming Error 2.1

When your `form` has several `checkboxes` with the same name, make sure that they have different `values`, or the web server scripts will not be able to distinguish them.



Common Programming Error 2.2

Not setting the `name` attributes of the `radio` buttons in a group to the same name is a logic error because it lets the user select all of the `radio` buttons at the same time.

HTML Forms Elements

- Type **file** allows the user to send a locally stored file as part of the form submission data.
- Type **tel** enables entering a telephone number.
 - To ensure that the user enters a phone number in a proper format, you must add a **pattern** attribute that uses a **regular expression** to determine whether the number is in the format.

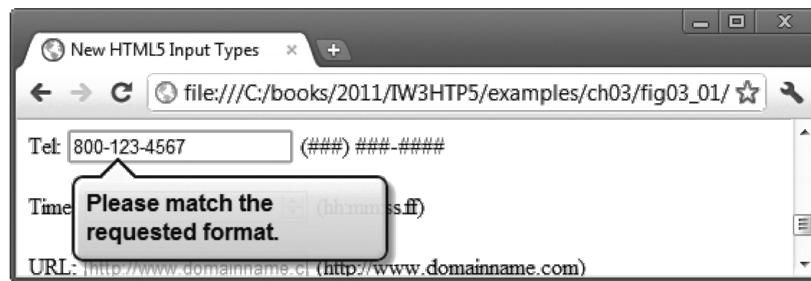


Fig. 3.14 | Validating a phone number using the **pattern** attribute in the **tel** input type.

HTML Forms Elements

- Type **password** inserts a password box into a form.
 - Allows users to enter sensitive information, such as credit card numbers and passwords, by “masking” the information input with another character, usually asterisks.
 - The actual value input is sent to the web server, not the asterisks that mask the input.
- Type **submit** creates a button that sends the form data to the location in the form’s action attribute.
- Type **reset** allows a user to reset all form elements to their default values.

Form Element's Attributes

- The **placeholder** attribute allows to place temporary text in a text field.
 - Generally, placeholder text is light gray provides an example of the text and/or text format the user should enter

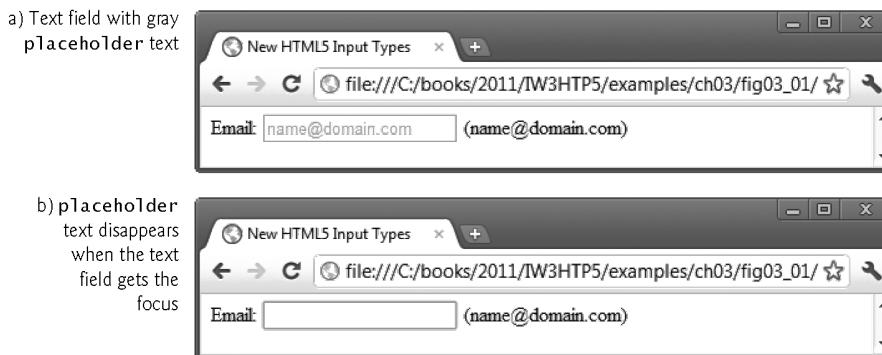


Fig. 3.8 | placeholder text disappears when the `input` element gets the focus.

Form Element's Attributes

- The **required** attribute forces the user to enter a value before submitting the form.

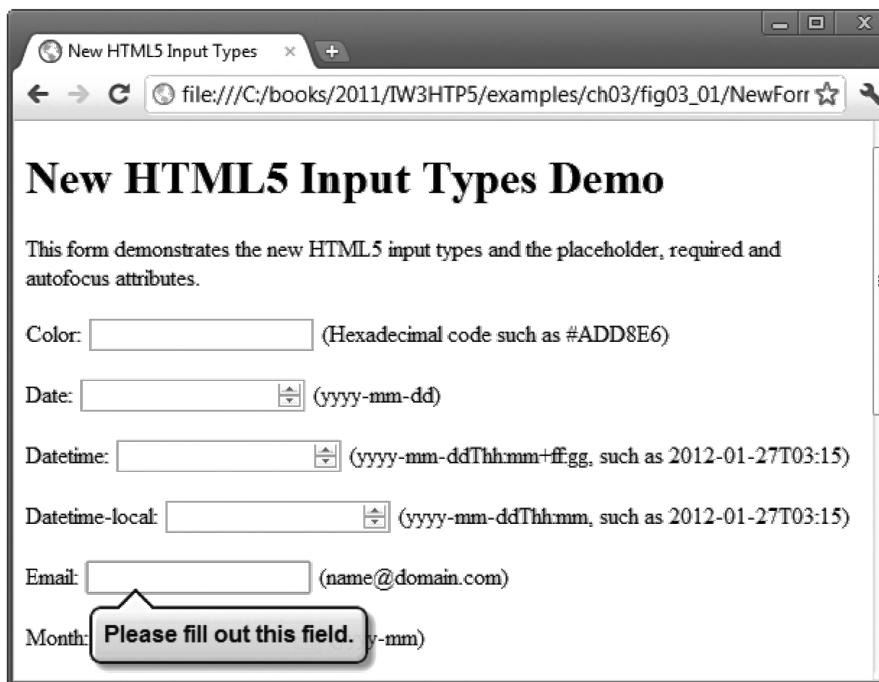


Fig. 3.9 | Demonstrating the required attribute in Chrome.

Form Element's Attributes

- The **autofocus** attribute specifies that an `<input>` element should automatically get focus when the page loads..
- The **minlength** attribute specifies the minimum number of characters required in an input field.
- The **maxlength** attribute specifies the maximum number of characters required in an input field.
- The **pattern** attribute specifies a regular expression that the `<input>` element's value is checked against on form submission.
- The **disabled** attribute disables an input element. It can be set to keep a user from using the element until some other condition has been met.

HTML Forms: Example

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.15: form2.html -->
4  <!-- Form using a variety of components. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>More Forms</title>
9      </head>
10
11     <body>
12         <h1>Feedback Form</h1>
13         <p>Please fill out this form to help
14             us improve our site.</p>
15
16         <form method = "post" action = "http://www.deitel.com">
17
18             <input type = "hidden" name = "recipient"
19                 value = "deitel@deitel.com">
20             <input type = "hidden" name = "subject"
21                 value = "Feedback Form">
22             <input type = "hidden" name = "redirect"
23                 value = "main.html">
24
```

Fig. 2.15 | Form using a variety of components. (Part I of 6.)

HTML Forms: Example

```
25 <p><label>Name:  
26     <input name = "name" type = "text" size = "25">  
27 </label></p>  
28  
29     <!-- <textarea> creates a multiline textbox -->  
30 <p><label>Comments:<br>  
31     <textarea name = "comments"  
32         rows = "4" cols = "36">Enter comments here.</textarea>  
33 </label></p>  
34  
35     <!-- <input type = "password"> inserts a -->  
36     <!-- textbox whose display is masked with -->  
37     <!-- asterisk characters -->  
38 <p><label>E-mail Address:  
39     <input name = "email" type = "password" size = "25">  
40 </label></p>  
41  
42 <p>  
43     <strong>Things you liked:</strong><br>  
44  
45     <label>Site design  
46         <input name = "thingsliked" type = "checkbox"  
47             value = "Design"></label>
```

Fig. 2.15 | Form using a variety of components. (Part 2 of 6.)

HTML Forms: Example

```
48 <label>Links
49   <input name = "thingsliked" type = "checkbox"
50     value = "Links"></label>
51 <label>Ease of use
52   <input name = "thingsliked" type = "checkbox"
53     value = "Ease"></label>
54 <label>Images
55   <input name = "thingsliked" type = "checkbox"
56     value = "Images"></label>
57 <label>Source code
58   <input name = "thingsliked" type = "checkbox"
59     value = "Code"></label>
60 </p>
61
62 <!-- <input type = "radio"> creates a radio -->
63 <!-- button. The difference between radio buttons -->
64 <!-- and checkboxes is that only one radio button -->
65 <!-- in a group can be selected. -->
66 <p>
67   <strong>How did you get to our site?</strong><br>
68
69   <label>Search engine
70     <input name = "howtosite" type = "radio"
71       value = "search engine" checked></label>
```

Fig. 2.15 | Form using a variety of components. (Part 3 of 6.)

HTML Forms: Example

```
72      <label>Links from another site
73          <input name = "howtosite" type = "radio"
74              value = "link"></label>
75      <label>Deitel.com Web site
76          <input name = "howtosite" type = "radio"
77              value = "deitel.com"></label>
78      <label>Reference in a book
79          <input name = "howtosite" type = "radio"
80              value = "book"></label>
81      <label>Other
82          <input name = "howtosite" type = "radio"
83              value = "other"></label>
84  </p>
85
86  <p>
87      <label>Rate our site:
88
89          <!-- the <select> tag presents a drop-down -->
90          <!-- list with choices indicated by the -->
91          <!-- <option> tags -->
92      <select name = "rating">
93          <option selected>Amazing</option>
94          <option>10</option>
95          <option>9</option>
96          <option>8</option>
```

Fig. 2.15 | Form using a variety of components. (Part 4 of 6.)

HTML Forms: Example

```
97      <option>7</option>
98      <option>6</option>
99      <option>5</option>
100     <option>4</option>
101     <option>3</option>
102     <option>2</option>
103     <option>1</option>
104     <option>Awful</option>
105   </select>
106 </label>
107 </p>
108
109 <p>
110   <input type = "submit" value = "Submit">
111   <input type = "reset" value = "Clear">
112 </p>
113 </form>
114 </body>
115 </html>
```

Fig. 2.15 | Form using a variety of components. (Part 5 of 6.)

HTML Forms: Example

The screenshot shows a feedback form titled "Feedback Form" displayed in a web browser window. The browser's title bar reads "More Forms" and the address bar shows the URL "file:///C:/books/2011/IW3HTP5/examples/ch02/form2.html". The form itself has the following components:

- Name:** A text input field.
- Comments:** A text area with placeholder text "Enter comments here."
- E-mail Address:** A text input field.
- Things you liked:** A list of items with checkboxes:
 - Site design
 - Links
 - Ease of use
 - Images
 - Source code
- How did you get to our site?:** A list of items with radio buttons:
 - Search engine
 - Links from another site
 - Deitel.com Web site
 - Reference in a book
 - Other
- Rate our site:** A dropdown menu currently set to "Amazing". Below it is a vertical list of numbers from 10 down to 1, with "C" next to the top item.
 - Submit
 - C 10
 - 9
 - 8
 - 7
 - 6
 - 5
 - 4
 - 3
 - 2
 - 1
 - Awful

Fig. 2.15 | Form using a variety of components. (Part 6 of 6.)

HTML Form Autocomplete

- **Autocomplete** is a feature that allows a user to start typing a value in a form field, and then the browser suggests possible values based on **previous entries**, **saved form data**, or a pre-defined list of options.
- This can help users **fill out forms faster** and **reduce errors** by suggesting or completing data that the user may not remember or know.

HTML Form Autocomplete

- The **autocomplete attribute** can be used on input types to automatically fill in the user's information based on previous input 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 **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.

HTML Form Autocomplete

- **Datalist** is an HTML element that provides a **pre-defined list of options** for an input field. It allows users to select from a list of suggested options as they type.
- The **alist element** can be useful for improving the **usability** of web forms by providing users with a list of suggested options to choose from, which can **save time** and **reduce errors**.

HTML Form Autocomplete

```
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 I of 6.)

HTML Form Autocomplete

```
23 <p><label>Last Name:<br/>
24   <input type = "text" id = "lastName"<br/>
25     placeholder = "Last name" /> (Last name)<br/>
26 </label></p>
27 <p><label>Email:<br/>
28   <input type = "email" id = "email"<br/>
29     placeholder = "name@domain.com" /> (name@domain.com)<br/>
30 </label></p>
31 <p><label for = "txtList">Birth Month:<br/>
32   <input type = "text" id = "txtList"<br/>
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.)

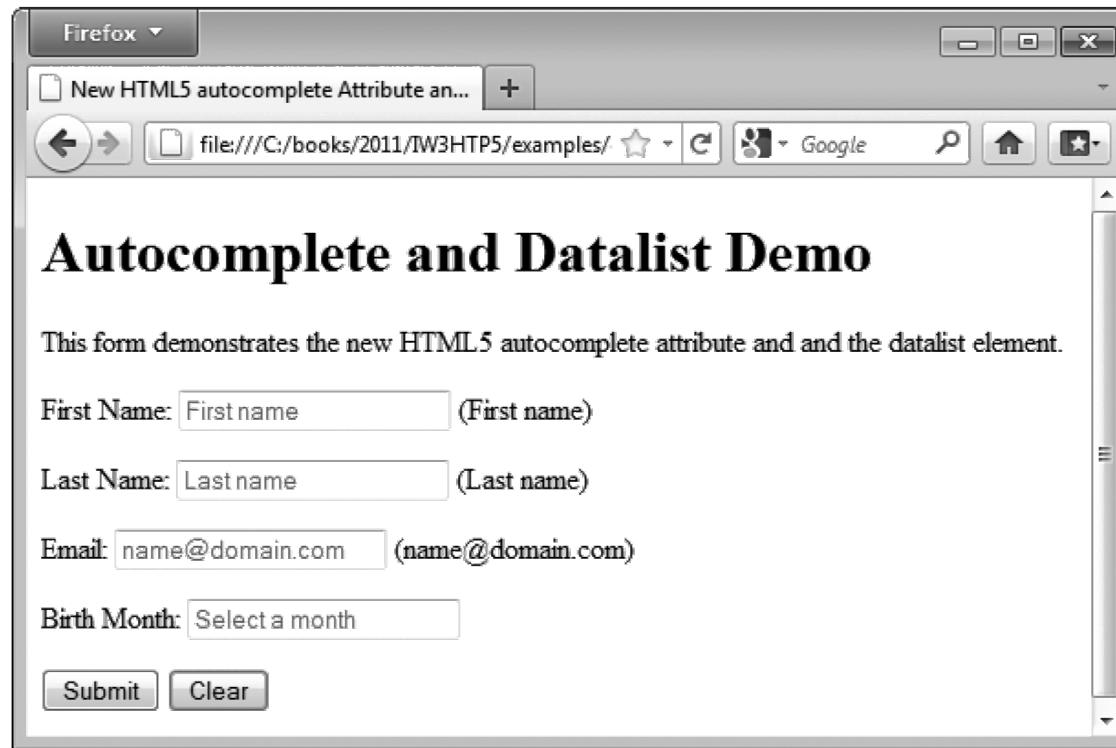
HTML Form Autocomplete

```
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.)

HTML Form Autocomplete

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



The screenshot shows a Firefox browser window with the title bar "Firefox" and a tab labeled "New HTML5 autocomplete Attribute an...". The address bar shows the URL "file:///C:/books/2011/TW3HTP5/examples/". The main content area displays the following HTML code:

```
<h2>Autocomplete and Datalist Demo</h2>


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:


```

The form fields are displayed with their respective placeholder text and descriptions in parentheses.

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

HTML Form Autocomplete

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

Firefox ▾

New HTML5 autocomplete Attribute an... +

file:///C:/books/2011/TW3HTP5/examples/ Google

Autocomplete and Datalist Demo

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

First Name: J (First name)
Jane

Last Name: Last name (Last name)

Email: name@domain.com (name@domain.com)

Birth Month: Select a month

Submit Clear

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

HTML Form Autocomplete

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

Firefox ▾

New HTML5 autocomplete Attribute an... +

file:///C:/books/2011/TW3HTP5/examples/ ⌂ Google

Autocomplete and Datalist Demo

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

First Name: Jane (First name)

Last Name: Blue (Last name)

Email: jane@domain.com (name@domain.com)

Birth Month: j

Submit

June
January
June
July

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

Form Data Validation

- **HTML data validation** is the process of checking that the data entered by a user into an HTML form **meets certain requirements or constraints** such as being of the correct **type, length, format**, or within a **specified range** of values.
- This is an important aspect of web development because it helps **ensure** the **quality, consistency**, and **security** of user input, as well as **prevent errors, data loss, or security breaches**.

Form Data Validation

- Data validation should be done on two layers:
 - **Client-side validation:** is performed in the user's web browser using HTML and JavaScript.
 - **Server-side validation:** is done on the web server using a scripting language .This can be done using many programming languages.
- The JavaScript data validation for client-side and server-side will be covered later in the course.

Form Data Validation

To create an **HTML validation layer**, we should: *client side*

- Set the '**required**' attribute when an input must be filled.
- Set the '**minlength**' and '**maxlength**' attributes for **all** text fields according to expected range.
- Set the '**pattern**' attribute for those elements that need to be in a specific **format**, e.g., the email or phone number.

Form Data Validation – Example 1

- **Task:** Create proper HTML **data validation** for a field that request a **mobile number** from a user. Number must be **9 digits** exactly.
- **Solution:**
 - **minlength** and **maxlength** were set to the specific range.
 - The **required** keyword used.
 - The **pattern** attribute used to accept only digits within range 0-9 and only accept 9 digits.

```
<input id="mobile" type='tel' required name="mobile" value="" placeholder="E.g.,  
05555550" pattern="[0-9]{9}" minlength="9" maxlength="9">
```

Form Data Validation – Example 2

- **Task:** Create proper HTML **data validation** for a field that requests an **email address** from a user with **length** of 3 to 150 characters.
- **Solution:**
 - **minlength** and **maxlength** were set to the specific range.
 - The **required** keyword used.
 - The **pattern** attribute used to accept email format.

```
<input id="email" type='email' name="email" value="" required  
minlength="3" maxlength="150" placeholder="E.g., m@m.com"  
pattern=" [a-zA-Z0-9_-]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,4}\.*[a-zA-Z]*  
">
```

<Any number of> [letters, numbers, underscore, dash] followed by <single> @ sign followed by <any number of> [letters, numbers] followed by <single> dot followed by <2 to 4> [letters]

HTML5 Semantic Tags

طريقة تنظيم

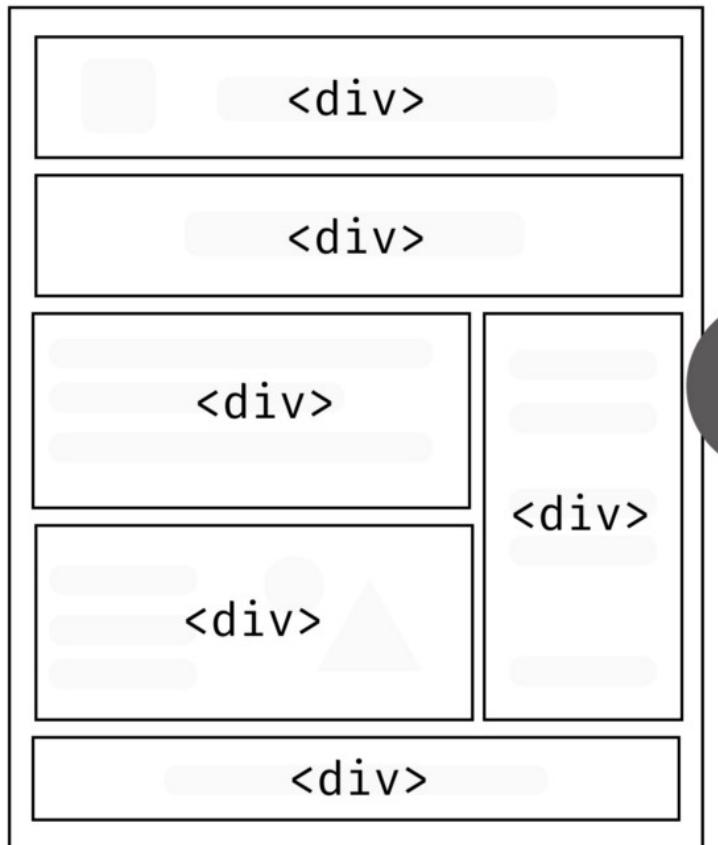
- HTML5 semantic tags are HTML elements that have a specific **meaning** and **purpose** in the structure and content of a web page.
- This include elements such as **header**, **nav**, **section**, **article**, **aside**, **footer**, **main**, **figure**, **figcaption**, and others.
- They were introduced in **HTML5** to provide a more **descriptive**, **meaningful**, and **accessible** way of **organizing** and presenting web content, especially for **search engines**, **screen readers**, and other **assistive technologies**.

HTML5 Semantic Tags

- HTML5 semantic tags are HTML elements that have a specific **meaning** and **purpose** in the structure and content of a web page.
- This include elements such as **header**, **nav**, **section**, **article**, **aside**, **footer**, **main**, **figure**, **figcaption**, and others.
- They were introduced in **HTML5** to provide a more **descriptive**, **meaningful**, and **accessible** way of **organizing** and presenting web content, especially for **search engines**, **screen readers**, and other **assistive technologies**.

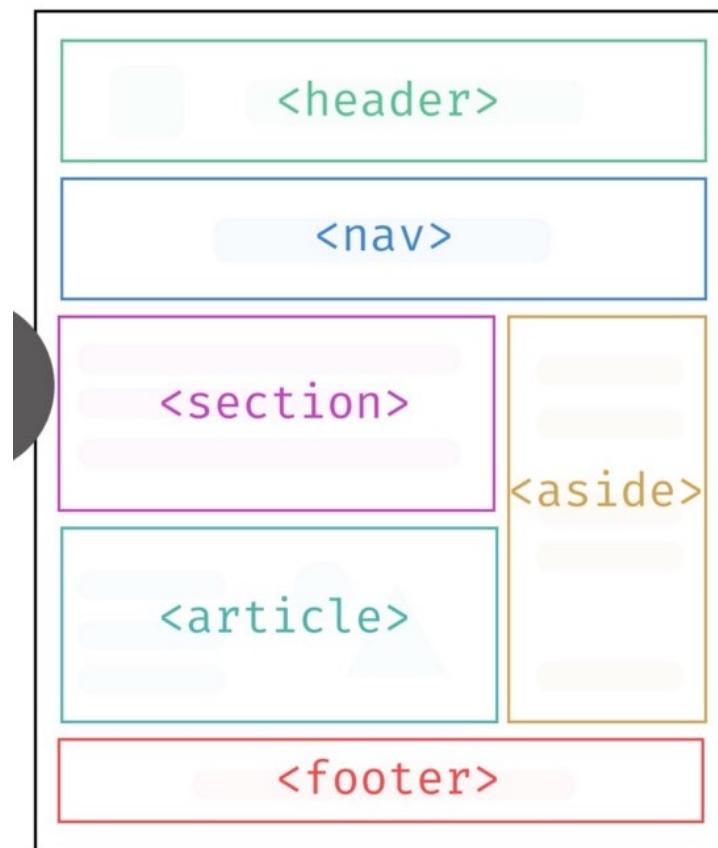
HTML5 Semantic Tags

Prior to HTML5



Ambiguous Sections

After HTML5



Clear Sections

HTML5 Semantic Tags

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 3.18: sectionelements.html --&gt;
4  &lt;!-- New HTML5 section elements. --&gt;
5  &lt;html&gt;
6      &lt;head&gt;
7          &lt;meta charset="utf-8"&gt;
8          &lt;title&gt;New HTML5 Section Elements&lt;/title&gt;
9      &lt;/head&gt;
10
11     &lt;body&gt;
12         &lt;header&gt; &lt;!-- header element creates a header for the page --&gt;
13             &lt;img src = "deitellogo.png" alt = "Deitel logo" /&gt;
14             &lt;h1&gt;Welcome to the Deitel Buzz Online&lt;h1&gt;
15
16             &lt;!-- time element inserts a date and/or time --&gt;
17             &lt;time&gt;2012-01-17&lt;/time&gt;
18
19         &lt;/header&gt;
20
21         &lt;section id = "1"&gt; &lt;!-- Begin section 1 --&gt;
22             &lt;nav&gt; &lt;!-- nav element groups navigation links --&gt;
23                 &lt;h2&gt; Recent Publications&lt;/h2&gt;</pre>
```

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

HTML5 Semantic Tags

```
24 <ul>
25   <li><a href = "http://www.deitel.com/books/iw3htp5">
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/jhtp9/">
33     Java How to Program, 9/e</a></li>
34   <li><a href = "http://www.deitel.com/books/cpphtp8/">
35     C++ How to Program, 8/e</a></li>
36   <li>
37     <a href = "http://www.deitel.com/books/vcsharp2010htp">
38       Visual C# 2010 How to Program, 4/e</a></li>
39   <li><a href = "http://www.deitel.com/books/vb2010htp">
40     Visual Basic 2010 How to Program</a></li>
41   </ul>
42 </nav>
43 </section>
44
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>
```

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

HTML5 Semantic Tags

```
48
49    <figure> <!-- figure element describes the image -->
50        <img src = "jhttp.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/jhttp9/">
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.)

HTML5 Semantic Tags

```
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 
```

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

HTML5 Semantic Tags

```
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>
```

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

HTML5 Semantic Tags

```
114      <li>"An easy-to-read conversational style.  
115          Clear code examples propel readers to  
116          become proficient in Java."  
117          <strong>&mdash;Patty Kraft, San Diego State  
118          University</strong></li>  
119      <li>"A great textbook with a myriad of examples  
120          from various application domains&mdash;  
121          excellent for a typical CS1 or CS2 course."  
122          <strong>&mdash;William E. Duncan, Louisiana  
123          State University</strong></li>  
124      </ul>  
125  </details>  
126  </p>  
127 </article>  
128  
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
```

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

HTML5 Semantic Tags

```
135 <h2>Deitel Developer Series Books</h2>
136 <h3><em>Android for Programmers: An App-Driven Approach
137 </em></h3>
138 Click <a href = "http://www.deitel.com/books/androidfp/">
139 here</a> for more information or to order this book.
140
141 <h2>LiveLessons Videos</h2>
142 <h3><em>C# 2010 Fundamentals LiveLessons</em></h3>
143 Click <a href = "http://www.deitel.com/Books/LiveLessons/">
144 here</a> for more information about our LiveLessons videos.
145 </section>
146
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
```

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

HTML5 Semantic Tags

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

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

HTML5 Semantic Tags

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.)

HTML5 Semantic Tags

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

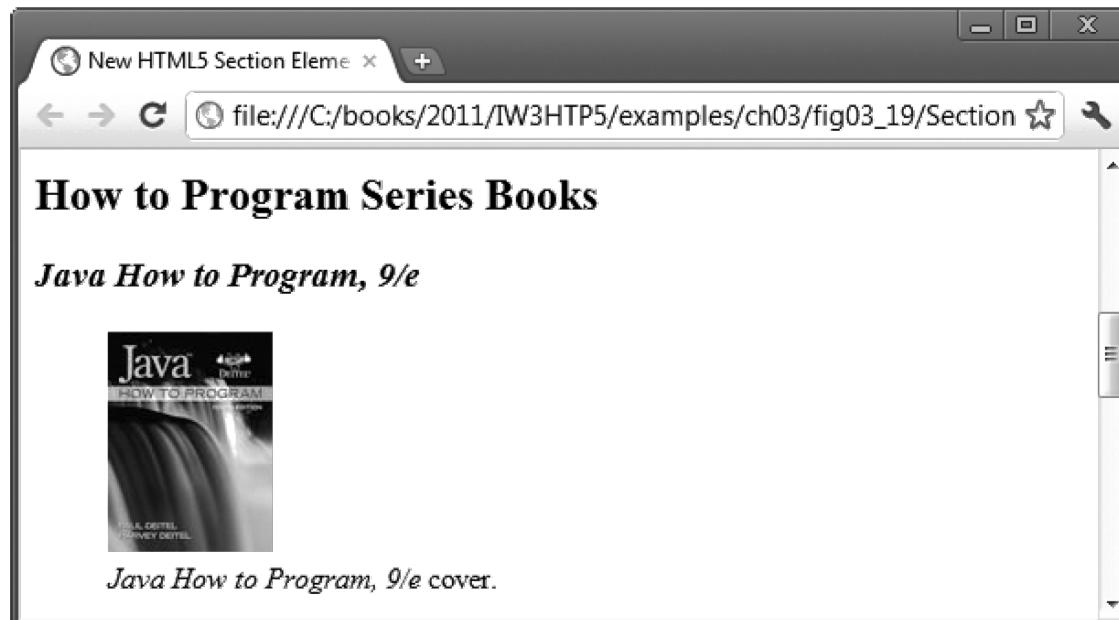


Fig. 3.18 | New HTML5 section elements. (Part 10 of 13.)

HTML5 Semantic Tags

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

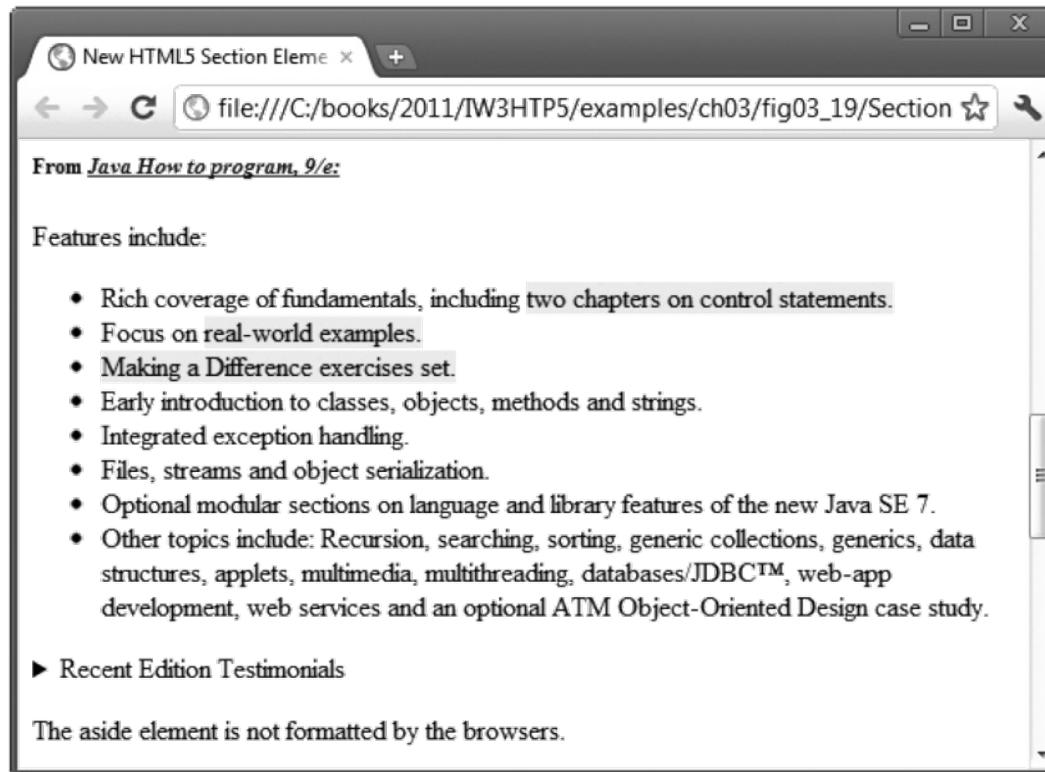


Fig. 3.18 | New HTML5 section elements. (Part 11 of 13.)

HTML5 Semantic Tags

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

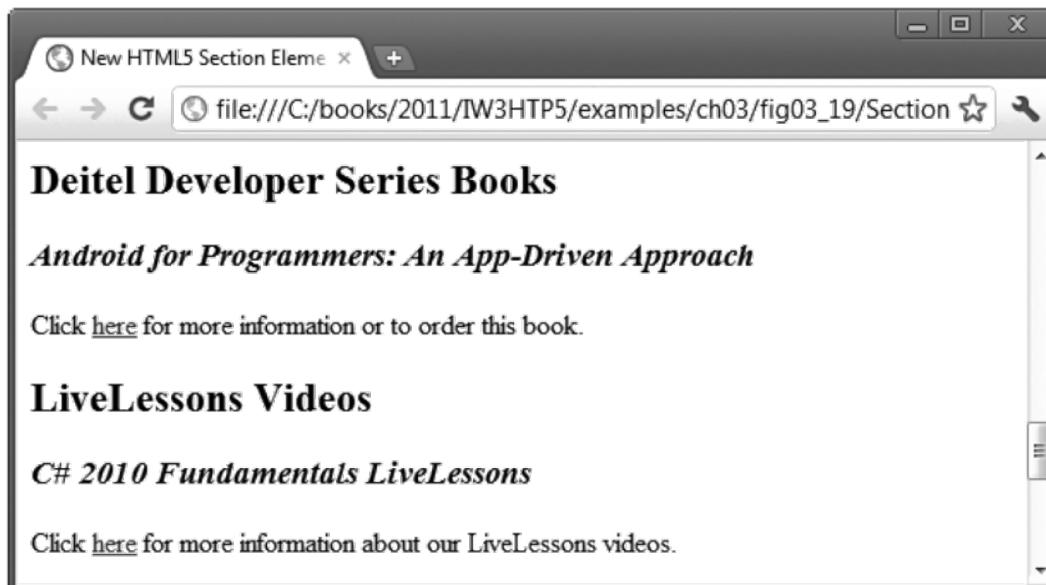


Fig. 3.18 | New HTML5 section elements. (Part 12 of 13.)

HTML5 Semantic Tags

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.)

What is Website Accessibility?

- **Website accessibility** is the practice of making websites **usable** by people of all abilities and disabilities.
- It involves designing and developing websites in a way that ensures **equal access** to information and functionality for all users, regardless of their abilities or disabilities.

Why is Website Accessibility important?

- There are millions of people with **disabilities** around the world who rely on the internet to access information, services, and products.
- Websites that are **inaccessible** can **prevent** people with disabilities from accessing important information and services, and can limit their ability to participate fully in society.
- In addition, **website accessibility** is a **legal requirement** in many countries, and failing to comply with accessibility standards can result in lawsuits and legal penalties.

Examples of Disabilities

- **Visual** impairments, such as blindness or low vision
- **Hearing** impairments, such as deafness or hard-of-hearing
- **Motor** impairments, such as paralysis or tremors
- **Cognitive** impairments, such as dyslexia or ADHD
- **Other** disabilities, such as color blindness or epilepsy

Website Accessibility Standards

- There are several **accessibility standards** that have been developed to ensure that websites are accessible to people with disabilities.
- The most widely used standard is the **Web Content Accessibility Guidelines (WCAG)**, which provides a set of guidelines for making websites accessible to people with disabilities.
- Other standards include Section 508, the Americans with Disabilities Act (ADA), and the **Accessible Rich Internet Applications (ARIA)** specification.

Techniques for Website Accessibility

- There are **several techniques** that can be used **to make websites accessible** to people with disabilities:
 - Providing **alternative text** for **images** and other non-text content
 - Ensuring that website functionality can be accessed using only the **keyboard**.
 - Using **HTML5 semantic tags** to organize content.
 - Using <**labels**> with form elements.
 - Using **high-contrast colors** and clear typography.
 - Providing **captions and transcripts** for videos and audio content.
 - Using **ARIA attributes** to improve accessibility for screen readers.

Website Accessibility: ARIA

- ARIA (Accessible Rich Internet Applications) is a **set of attributes** that can be **added to HTML elements** to improve the accessibility of web content for users with disabilities, particularly those who use assistive technologies.
- **ARIA attributes** provide **additional** information about the **purpose, function, and state** of **elements** on a web page that may **not be readily apparent** from the visual presentation alone.

Website Accessibility: ARIA

- **ARIA Roles** define the purpose and structure of an element. For example, stating role as a button, checkbox, menu, dialog, etc.
- **ARIA States and Properties** describe the current state or value of an element. For example, `aria-checked`, `aria-disabled`, `aria-haspopup`, `aria-selected`, etc.
- **ARIA Labels and Descriptions** provide additional context and information about an element. For example, `aria-label`, `aria-labelledby`, `aria-describedby`.

Website Accessibility: ARIA

- ARIA attributes **do not modify** the visual appearance or behavior of an element on their own. Only **users who rely on assistive technologies** will notice the differences between a digital product with ARIA and one without it.
- For more details on ARIA and its specifications, you can refer to the W3C document <https://www.w3.org/TR/html-aria/>

Testing Website Accessibility

- It's important to test your website for accessibility to ensure that it is usable by people with disabilities.
- There are several tools available for **testing website accessibility**, including:
 - Web Accessibility Evaluation Tools, such as the WAVE tool and the Axe tool
 - Screen readers, such as JAWS and NVDA
 - User testing with people with disabilities



Any questions?
Please feel free to raise your
hands and ask.