

Chapter 2

Introduction to HTML5

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

Topics

- Tables
 - Forms
 - Internal Linking
 - Meta* Elements
-

2.10 Tables

- Tables are frequently used to organize data into *rows* and *columns*.
- The **table** element defines an HTML5 table
- 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.
- The **caption** element specifies a table's title.
- It's good practice to 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.

```
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
```

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

```
20      <!-- the <thead> section appears first in the table -->
21      <!-- it formats the table header area -->
22      <thead>
23          <tr> <!-- <tr> inserts a table row -->
24              <th>Fruit</th> <!-- insert a heading cell -->
25              <th>Price</th>
26          </tr>
27      </thead>
28
29      <!-- the <tfoot> section appears last in the table -->
30      <!-- it formats the table footer -->
31      <tfoot>
32          <tr>
33              <th>Total</th>
34              <th>$3.75</th>
35          </tr>
36      </tfoot>
37
38      <!-- all table content is enclosed -->
39      <!-- within the <tbody> -->
40      <tbody>
41          <tr>
42              <td>Apple</td> <!-- insert a data cell -->
43              <td>$0.25</td>
44          </tr>
```

Fig. 2.12 | Creating a basic table. (Part 2 of 4.)

```
45      <tr>
46          <td>Orange</td>
47          <td>$0.50</td>
48      </tr>
49      <tr>
50          <td>Banana</td>
51          <td>$1.00</td>
52      </tr>
53      <tr>
54          <td>Pineapple</td>
55          <td>$2.00</td>
56      </tr>
57  </tbody>
58 </table>
59 </body>
60 </html>
```

Fig. 2.12 | Creating a basic table. (Part 3 of 4.)

A screenshot of a web browser window titled "A simple HTML5 table". The browser interface includes standard navigation buttons (back, forward, search, etc.) and a toolbar with links, bookmarks, and sync error indicators. The main content area displays a table with the following structure:

Table caption: **Table of Fruits
(1st column) and
Their Prices
(2nd column)**

Table header: **Fruit** | **Price**

Table body:
Apple | \$0.25
Orange | \$0.50
Banana | \$1.00
Pineapple | \$2.00

Table footer: **Total** | **\$3.75**

The table has a single border around the entire structure.

Fruit	Price
Apple	\$0.25
Orange	\$0.50
Banana	\$1.00
Pineapple	\$2.00
Total	\$3.75

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

2.10 Tables (Cont.)

- ❑ A table can be split into three distinct sections:
 - Head (**thead** element)
 - ❑ Table titles
 - ❑ Column headers
 - 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

2.10 Tables (Cont.)

❑ tr Element

- Defines individual table rows
- Element th
 - ❑ Defines a header cell

❑ Td Element

- Contains table data elements

2.10 Tables (Cont.)

Using rowspan and colspan with Tables

- Figure 2.13 introduces two new attributes that allow you to build more complex tables.
- You can merge data cells with the `rowspan` and `colspan` attributes
 - The values of these attributes specify the number of rows or columns occupied by the cell.
 - Can be placed inside any data cell or table header cell.
- The `br` element is rendered as a line break in most browsers—any markup or text following a `br` element is rendered on the next line.
 - Like the `img` element, `br` is an example of a void element.
 - Like the `hr` element, `br` is considered a legacy formatting element that you should avoid using—in general, formatting should be specified using CSS.

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.13: table2.html -->
4  <!-- Complex HTML5 table. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Tables</title>
9      </head>
10
11     <body>
12         <h1>Table Example: Spanning Rows and Columns</h1>
13
14         <table border = "1">
15             <caption>A more complex sample table</caption>
16
```

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

EXAMPLE-Cont...

```
17 <thead>
18     <!-- rowspans and colspans merge the specified -->
19     <!-- number of cells vertically or horizontally -->
20     <tr>
21         <!-- merge two rows -->
22         <th rowspan = "2">
23             <img src = "camel.png" width = "205"
24                 height = "167" alt = "Picture of a camel">
25         </th>
26
27         <!-- merge four columns -->
28         <th colspan = "4">
29             <strong>Camelid comparison</strong><br>
30             Approximate as of 10/2011
31         </th>
32     </tr>
33     <tr>
34         <th># of humps</th>
35         <th>Indigenous region</th>
36         <th>Spits?</th>
37         <th>Produces wool?</th>
38     </tr>
39 </thead>
```

Fig. 2.13 | Complex HTML5 table. (Part 2 of 4.)

EXAMPLE-Cont...

```
40      <tbody>
41          <tr>
42              <th>Camels (bactrian)</th>
43              <td>2</td>
44              <td>Africa/Asia</td>
45              <td>Yes</td>
46              <td>Yes</td>
47      </tr>
48      <tr>
49          <th>Llamas</th>
50          <td>1</td>
51          <td>Andes Mountains</td>
52          <td>Yes</td>
53          <td>Yes</td>
54      </tr>
55  </tbody>
56 </table>
57 </body>
58 </html>
```

Fig. 2.13 | Complex HTML5 table. (Part 3 of 4.)

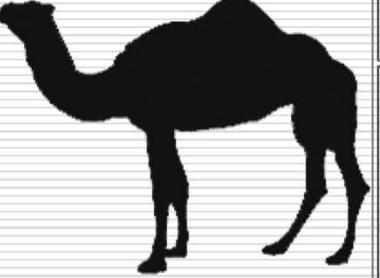
EXAMPLE-Output

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.)

2.11 Forms

- HTML5 provides **forms** for collecting information from users.
- Figure 2.14 is a simple form that sends data to the web server for processing.

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.14: form.html -->
4  <!-- Form with a text field and hidden fields. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Forms</title>
9      </head>
10
11     <body>
12         <h1>Feedback Form</h1>
13
14         <p>Please fill out this form to help
15             us improve our site.</p>
16
17         <!-- this tag starts the the form, gives the -->
18         <!-- method of sending information and the -->
19         <!-- location of the form-processing script -->
20         <form method = "post" action = "http://www.deitel.com">
21             <!-- hidden inputs contain non-visual -->
22             <!-- information that will also be submitted -->
23             <input type = "hidden" name = "recipient"
24                 value = "deitel@deitel.com">
```

Fig. 2.14 | Form with a text field and hidden fields. (Part I of 3.)

EXAMPLE (Cont...)

```
25      <input type = "hidden" name = "subject"
26          value = "Feedback Form">
27      <input type = "hidden" name = "redirect"
28          value = "main.html">
29
30      <!-- <input type = "text"> inserts a text field -->
31      <p><label>Name:
32          <input name = "name" type = "text" size = "25"
33              maxLength = "30">
34      </label></p>
35
36      <p>
37          <!-- input types "submit" and "reset" insert -->
38          <!-- buttons for submitting and clearing the -->
39          <!-- form's contents, respectively -->
40          <input type = "submit" value = "Submit">
41          <input type = "reset" value = "Clear">
42      </p>
43      </form>
44  </body>
45 </html>
```

Fig. 2.14 | Form with a text field and hidden fields. (Part 2 of 3.)

EXAMPLE-Output

The screenshot shows a web browser window titled "Forms". The address bar displays the URL "file:///C:/books/2011/IW3HTP5/examples/ch02/form.html". The main content area is titled "Feedback Form" in large, bold, black font. Below the title, a message reads "Please fill out this form to help us improve our site." A text input field is labeled "Name:" followed by a blank rectangular box. At the bottom of the form are two buttons: "Submit" and "Clear". The browser interface includes standard controls like back, forward, and search at the top.

Fig. 2.14 | Form with a text field and hidden fields. (Part 3 of 3.)

2.11 Forms (Cont.)

method Attribute of the form Element

- A form is defined by a **form** element
 - 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.
 - 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.
 - The **action** attribute of the form element specifies the script to which the form data will be sent

2.11 Forms (Cont.)

action Attribute of the form Element

- The **action** attribute of the form element specifies the script to which the form data will be sent.
- Since we haven't introduced server-side programming yet, we set this attribute to `http://www.deitel.com` for now.
- **input** elements that specify data to provide to the script that processes the form (also called the **form handler**).
- An input's type is determined by its **type attribute**.

2.11 Forms (Cont.)

Hidden Inputs

- ❑ 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.

2.11 Forms (Cont.)

text input Element

- The **text input** inserts a text field into the form, which allows the user to input data.
- The **label** element provides users with information about the input element's purpose
- The **size** attribute specifies the number of characters visible in the text field.
- Optional attribute **maxlength** limits the number of characters input into a text field.

2.11 Forms (Cont.)

submit and reset input Elements

- The **submit** input element is a button.
 - When the submit button is pressed, the form's data is sent to the location specified in the form's action attribute.
- The **value** attribute sets the text displayed on the button.
- The **reset** input element allows a user to reset all form elements to their default values.

2.11 Forms (Cont.)

Additional Form Elements

- ❑ Figure 2.15 contains a form that solicits user feedback about a website.
- ❑ The **textarea** element inserts a *multiline text area* into the form.
- ❑ The number of rows is specified with the **rows** attribute, and the number of columns (i.e., characters per line) with the **cols** attribute.
- ❑ Default text can be specified in other **input** types, such as text fields, by using the **value** attribute.

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.)

EXAMPLE (Cont...)

```
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.)

EXAMPLE (Cont...)

```
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.)

EXAMPLE (Cont...)

```
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.)

EXAMPLE (Cont...)

```
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.)

EXAMPLE-Output

The screenshot shows a window titled "More Forms" with the URL "file:///C:/books/2011/IW3HTP5/examples/ch02/form2.html". The form is titled "Feedback Form" and contains the following fields:

- Name:** An input field with the placeholder "John Doe".
- Comments:** A text area with the placeholder "Enter comments here." containing the text "I like your site".
- E-mail Address:** An input field with the placeholder "johndoe@example.com".
- Things you liked:** A list of checkboxes:
 - Site design
 - Links
 - Ease of use
 - Images
 - Source code
- How did you get to our site?:** A list of 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". The dropdown menu also includes options 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, and Awful.
- Submit:** A button labeled "Submit".

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

2.11 Forms (Cont.)

- The **password** input 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.

2.11 Forms (Cont.)

- The **checkbox** input element enables users to select and option.
 - When the **checkbox** is selected, a check mark appears in the **checkbox**. Otherwise, the checkbox is empty
 - checkboxes can be used individually and in groups. checkboxes that are part of the same group have the same name
- **radio** buttons are similar to **checkboxes**, except that only one radio button in a group can be selected at any time.
 - All radio buttons in a group have the same name attribute but different value attributes.
- The **select** input provides a drop-down list of items.
 - The **name** attribute identifies the drop-down list.
 - The **option** element adds items to the drop-down list.



Common Programming Error 2.1

When your form has several `checkbox`s 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.



2.12 Internal Linking

- The `a` tag can be used to link to another section of the same document by specifying the element's `id` as the link's `href`.
- To link internally to an element with its `id` attribute set, use the syntax `#id`.

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.16: internal.html -->
4  <!-- Internal Linking -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Internal Links</title>
9      </head>
10
11     <body>
12         <!-- id attribute creates an internal hyperlink destination -->
13         <h1 id = "features">The Best Features of the Internet</h1>
14
15         <!-- an internal link's address is "#id" -->
16         <p><a href = "#bugs">Go to <em>Favorite Bugs</em></a></p>
17
18         <ul>
19             <li>You can meet people from countries
20                 around the world.</li>
21             <li>You have access to new media as it becomes public:
22                 <ul>
23                     <li>New games</li>
```

Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part I of 5.)

EXAMPLE

```
24      <li>New applications
25          <ul>
26              <li>For Business</li>
27                  <li>For Pleasure</li>
28          </ul>
29      </li>
30
31      <li>Around the clock news</li>
32          <li>Search Engines</li>
33          <li>Shopping</li>
34          <li>Programming
35              <ul>
36                  <li>HTML5</li>
37                  <li>Java</li>
38                  <li>Dynamic HTML</li>
39                  <li>Scripts</li>
40                  <li>New languages</li>
41          </ul>
42      </li>
43      </ul>
44  </li>
45
```

Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 2 of 5.)

EXAMPLE (Cont...)

```
46      <li>Links</li>
47      <li>Keeping in touch with old friends</li>
48      <li>It is the technology of the future!</li>
49  </ul>
50
51  <!-- id attribute creates an internal hyperlink destination -->
52  <h1 id = "bugs">My 3 Favorite Bugs</h1>
53  <p>
54      <!-- internal hyperlink to features -->
55      <a href = "#features">Go to <em>Favorite Features</em></a>
56  </p>
57  <ol>
58      <li>Fire Fly</li>
59      <li>Gal Ant</li>
60      <li>Roman Tic</li>
61  </ol>
62  </body>
63 </html>
```

Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 3 of 5.)

EXAMPLE- Output

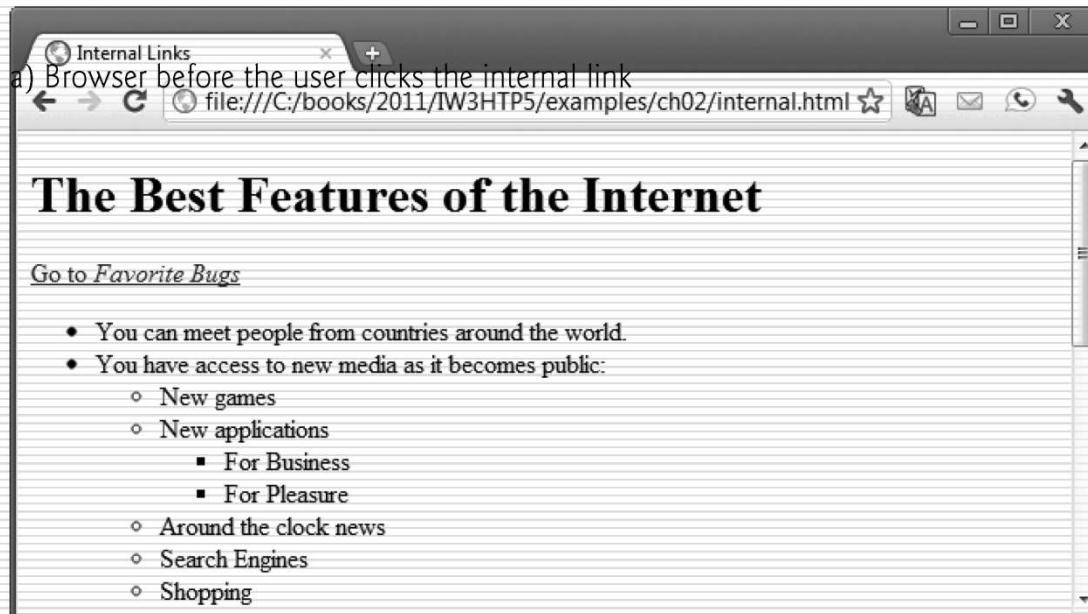


Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 4 of 5.)

EXAMPLE-Output

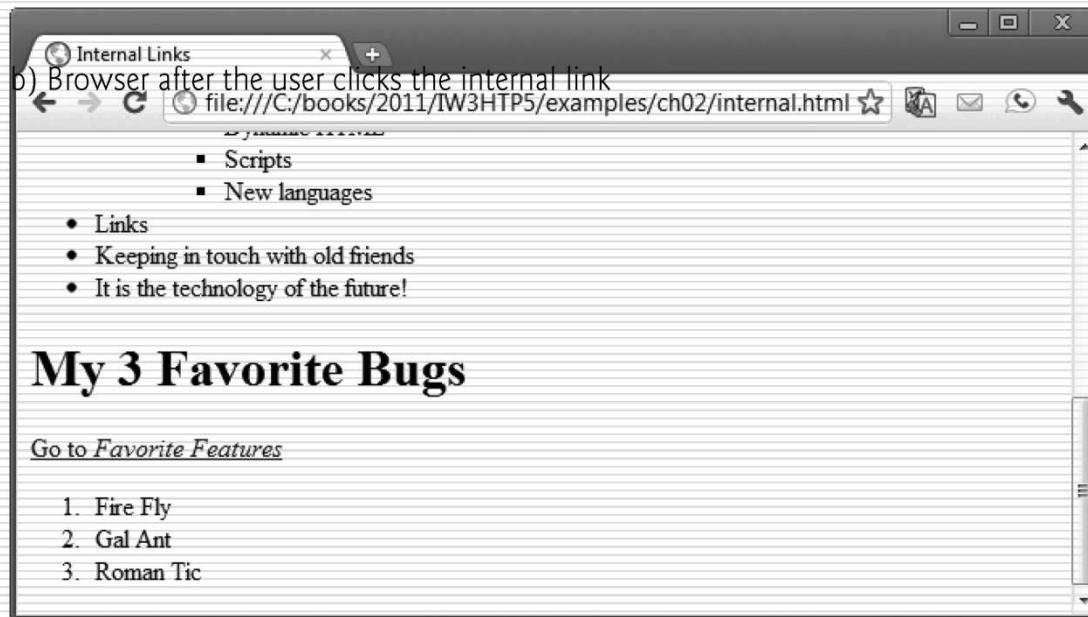


Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 5 of 5.)

2.13 meta Elements

- One way that search engines catalog pages is by reading the **meta** element's contents.
 - The **name** attribute identifies the type of **meta** element
 - The **content** attribute
 - Of a **keywords** **meta** element: provides search engines with a list of words that describe a page, which are compared with words in search requests
 - Of a **description** **meta** element: provides a three- to four-line description of a site in sentence form, used by search engines to catalog your site. This text is sometimes displayed as part of the search result

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.17: meta.html -->
4  <!-- meta elements provide keywords and a description of a page. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Welcome</title>
9
10         <!-- <meta> tags provide search engines with -->
11         <!-- information used to catalog a site -->
12         <meta name = "keywords" content = "web page, design,
13             HTML5, tutorial, personal, help, index, form,
14             contact, feedback, list, links, deitel">
15         <meta name = "description" content = "This website will
16             help you learn the basics of HTML5 and web page design
17             through the use of interactive examples and
18             instruction.">
19     </head>
```

Fig. 2.17 | meta elements provide keywords and a description of a page. (Part 1 of 3.)

EXAMPLE (Cont...)

```
20  <body>
21      <h1>Welcome to Our Website!</h1>
22
23      <p>We have designed this site to teach about the wonders
24          of <strong><em>HTML5</em></strong>. <em>HTML5</em> is
25          better equipped than <em>HTML</em> to represent complex
26          data on the Internet. <em>HTML5</em> takes advantage of
27          XML's strict syntax to ensure well-formedness. Soon you
28          will know about many of the great features of
29          <em>HTML5.</em></p>
30
31      <p>Have Fun With the Site!</p>
32  </body>
33 </html>
```

Fig. 2.17 | meta elements provide keywords and a description of a page. (Part 2 of 3.)

EXAMPLE (Cont...)

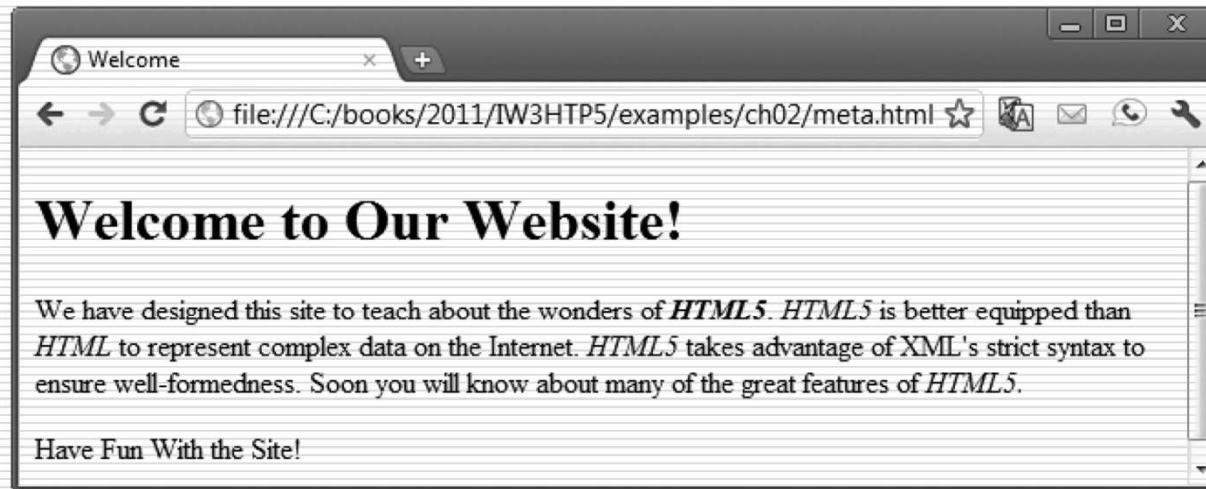


Fig. 2.17 | meta elements provide keywords and a description of a page. (Part 3 of 3.)



Software Engineering Observation 2.2

meta elements are not visible to users. They must be placed inside the head section of your HTML5 document; otherwise they will not be read by search engines.