

Web Database Programming

- **Dr. Abu Sayed Md. Mostafizur Rahaman**
- **Professor**
- Department of Computer Science and Engineering
- Jahangirnagar University

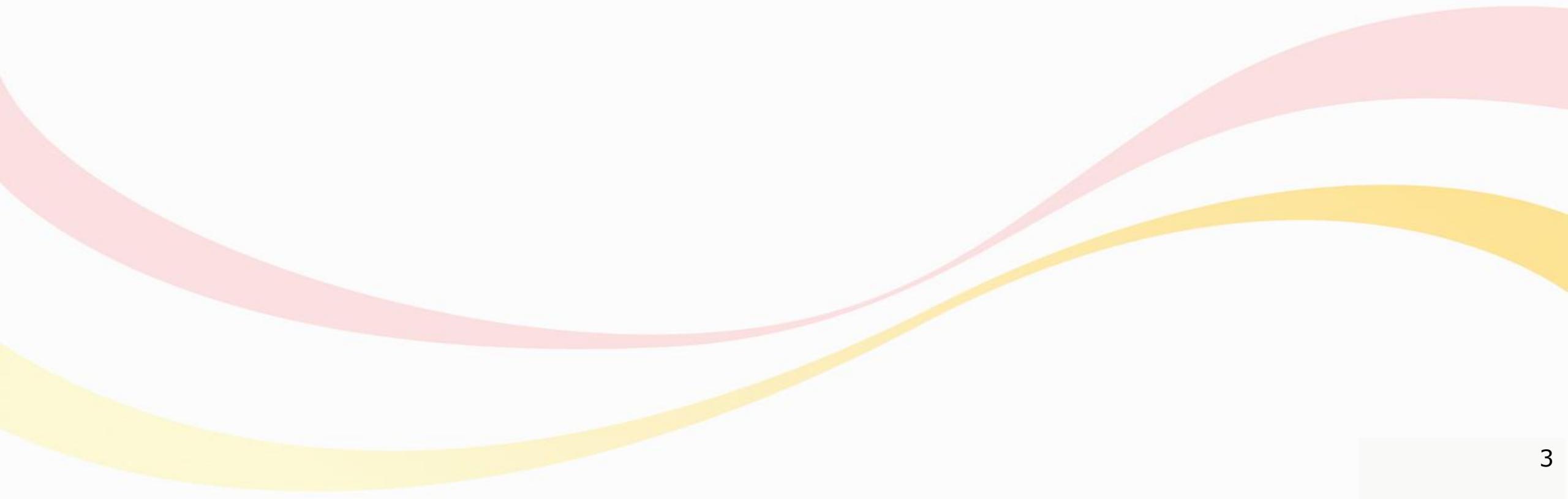
References

- Internet and World Wide Web How To Program (5th Edition) 5th Edition
 - by [Harvey & Paul\) Deitel & Associates](#), [Harvey Deitel](#), [Abbey Deitel](#)
- *Web Programming Step by Step*
 - by Marty Stepp, Jessica Miller, and Victoria Kirst
- <https://www.w3schools.com/>
- Pro Spring Boot by Gutierrez, Felipe
- Mastering Spring Boot 2.0 by Dinesh Rajput
- Spring Microservices in Action by John Carnell
- Spring Boot in Action by Craig Walls

Chapter 2

Introduction to HTML5

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



Topics

- Introduction
- Editing HTML5
- First HTML Example
- W3C HTML Validation Service
- Heading
- Linking
- Images
- Special Characters
- List

2.1 Introduction

- HTML5 (HyperText Markup Language 5)
 - markup language that specifies
 - the *structure* and
 - *content* of documents that are displayed in web browsers
- We introduce some basics, then cover more sophisticated HTML5 techniques such as:
 - **tables**, which are particularly useful for structuring information from databases (i.e., software that stores structured sets of data)
 - **forms** for collecting information from web-page visitors
 - **internal linking** for easier page navigation
 - **meta elements** for specifying information about a document

2.3 First HTML5 Example

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.1: main.html -->
4  <!-- First HTML5 example. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Welcome</title>
9      </head>
10
11     <body>
12         <p>Welcome to HTML5!</p>
13     </body>
14 </html>
```

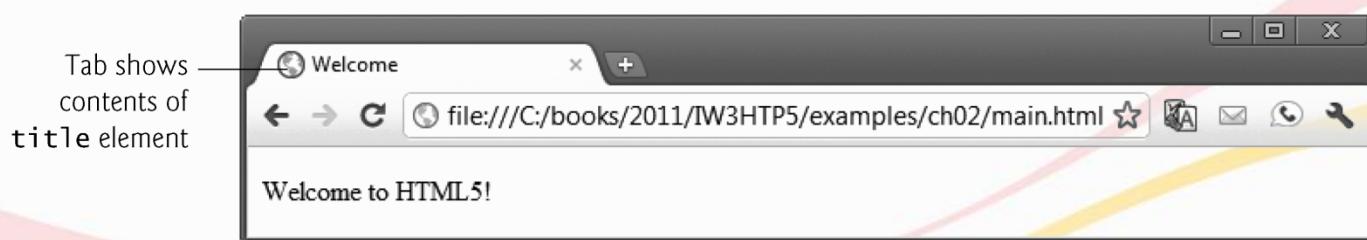


Fig. 2.1 | First HTML5 example.

2.3 First HTML5 Example

Document Type Declaration

- The **document type declaration (DOCTYPE)** is required in HTML5 documents so that browsers render the page in standards mode.

2.3 First HTML5 Example

Comments

- Insert comments in your HTML5 markup to improve readability and describe the content of a document.
- The browser ignores comments when your document is rendered.
- Comments start with **<!--** and end with **-->**.

2.3 First HTML5 Example

html, head and body Elements

- HTML5 markup contains text (and images, graphics, animations, audios and videos) that represents the content of a document and elements that specify a document's *structure and meaning*.
 - The **html** element *encloses* the head section (represented by the head element) and the body section (represented by the body element).
 - The **head section** contains information about the HTML5 document, such as the character set (UTF-8, the most popular character-encoding scheme for the web) that the page use—which helps the browser determine how to render the content—and the **title**.
 - The **head** section also can contain special document-formatting instructions called **CSS3 style sheets** and client-side programs called scripts for creating dynamic web pages.
 - The **body section** contains the page's content, which the browser displays when the user visits the web page.

2.3 First HTML5 Example (cont.)

Start Tags and End Tags

- HTML5 documents *delimit* most elements with a start tag and end tag.
- A **start tag** consists of the element name in **angle brackets**
 - For example, `<html>`
- An **end tag** consists of the element name preceded by a forward slash (/) in angle brackets
 - For example, `</html>`
- There are several so-called “void elements” that do not have end tags.
- Many start tags have **attributes** that provide additional information about an element, which browsers use to determine how to process the element.
- Each **attribute** has a **name** and a **value** separated by an equals sign (=).

2.3 First HTML5 Example (Cont.)

Title Element

- The **title** element is called a **nested element**, because it's enclosed in the **head** element's start and end tags.
- The **head** element is also a nested element, because it's enclosed in the **html** element's start and end tags.
- The **title** element describes the web page.
 - Titles usually appear in the title bar at the top of the browser window, in the browser tab on which the page is displayed, and also as the text identifying a page when users add the page to their list of Favorites or Bookmarks, enabling them to return to their favorite sites.
 - Search engines use the title for indexing purposes and when displaying results

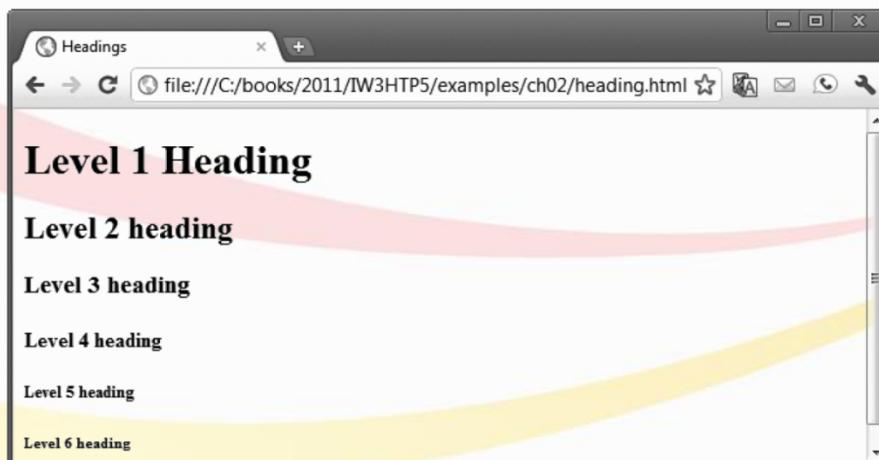
2.3 First HTML5 Example (Cont.)

Paragraph Element (<p> . . . </p>)

- All text placed between the `<p>` and `</p>` tags forms one paragraph.

2.5 Headings

- HTML5 provides six heading elements (`h1` through `h6`) for specifying the *relative importance* of information
 - Heading element **h1** is considered the most significant heading and is rendered in the largest font.
 - Each successive heading element (i.e., `h2`, `h3`, etc.) is rendered in a progressively smaller font.



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.2: heading.html -->
4  <!-- Heading elements h1 through h6. -->
5  <html>
6    <head>
7      <meta charset = "utf-8">
8      <title>Headings</title>
9    </head>
10   <body>
11     <h1>Level 1 Heading</h1>
12     <h2>Level 2 heading</h2>
13     <h3>Level 3 heading</h3>
14     <h4>Level 4 heading</h4>
15     <h5>Level 5 heading</h5>
16     <h6>Level 6 heading</h6>
17   </body>
18 </html>
```

Fig. 2.2 | Heading elements `h1` through `h6`. (Part 1 of 2.)

2.6 Linking

- A hyperlink references or links to other resources, such as HTML5 documents and images.
- Web browsers typically ***underline*** text hyperlinks and color them ***blue*** by default.

2.6 Linking

- A hyperlink references or links to other resources, such as HTML5 documents and images.
- Web browsers typically ***underline*** text hyperlinks and color them ***blue*** by default.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.3: Links.html -->
4  <!-- Linking to other web pages. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Links</title>
9      </head>
10
11     <body>
12         <h1>Here are my favorite sites:</h1>
13         <p><strong>Click a name to visit that site.</strong></p>
14
15         <!-- create four text hyperlinks -->
16         <p><a href = "http://www.facebook.com">Facebook</a></p>
17         <p><a href = "http://www.twitter.com">Twitter</a></p>
18         <p><a href = "http://www.foursquare.com">Foursquare</a></p>
19         <p><a href = "http://www.google.com">Google</a></p>
20     </body>
21 </html>
```

Fig. 2.3 | Linking to other web pages. (Part I of 2.)

EXAMPLE-Output



Fig. 2.3 | Linking to other web pages. (Part 2 of 2.)

2.6 Linking (Cont.)

- The **strong element** indicates that the content has high importance. Browsers typically render such text in a bold font.
- Links are created using the **a (anchor) element**.
- Attribute **href (hypertext reference)** specifies a resource's location, such as
 - a web page or location within a web page
 - a file
 - an e-mail address
- When a URL does not indicate a specific document on the website, the web server returns a default web page. This page is often called `index.html`, but most web servers can be configured to use any file as the default web page for the site.
- If the web server cannot locate a requested document, it returns an error indication to the web browser (known as a `404` error), and the browser displays a web page containing an error message.

2.6 Linking (Cont.)

Hyperlinking to an E-Mail Address

- Anchors can link to an e-mail address using a `mailto:` URL
 - When a user clicks this type of anchored link, most browsers launch the default e-mail program (e.g., Mozilla Thunderbird, Microsoft Outlook or Apple Mail) to enable the user to write an e-mail message to the linked address.

EXAMPLE

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.4: contact.html -->
4 <!-- Linking to an e-mail address. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Contact Page</title>
9   </head>
10
11  <body>
12    <p>
13      To write to <a href = "mailto:deitel@deitel.com">
14        Deitel & Associates, Inc.</a>, click the link and your default
15        email client will open an email message and address it to us.
16    </p>
17  </body>
18 </html>
```

Fig. 2.4 | Linking to an e-mail address. (Part 1 of 3.)

EXAMPLE-Output



Fig. 2.4 | Linking to an e-mail address. (Part 2 of 3.)

2.7 Images

- The most popular image formats used by web developers today are **PNG** (Portable Network Graphics) and **JPEG** (Joint Photographic Experts Group).
- Users can create images using specialized software, such as Adobe Photoshop Express (www.photoshop.com), G.I.M.P. (www.gimp.org), Inkscape (www.inkscape.org) and many more.
- Images may also be acquired from various websites, many of which offer royalty-free images.

EXAMPLE

```
1 <!DOCTYPE html>
2
3 <!-- Fig. 2.6: picture.html -->
4 <!-- Including images in HTML5 files. -->
5 <html>
6   <head>
7     <meta charset = "utf-8">
8     <title>Images</title>
9   </head>
10
11  <body>
12    <p>
13      <img src = "cpphttp.png" width = "92" height = "120"
14        alt = "C++ How to Program book cover">
15      <img src = "jhttp.png" width = "92" height = "120"
16        alt = "Java How to Program book cover">
17    </p>
18  </body>
19 </html>
```

Fig. 2.6 | Including images in HTML5 files. (Part 1 of 2.)

EXAMPLE-Output

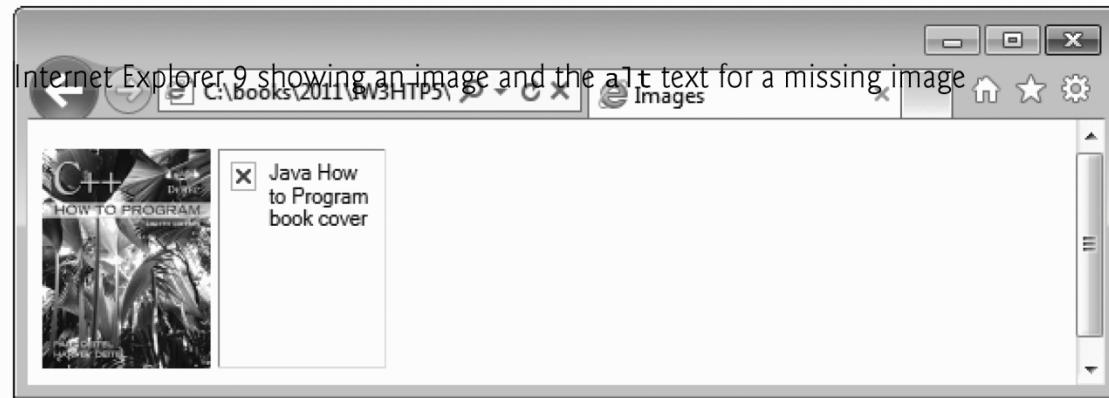


Fig. 2.6 | Including images in HTML5 files. (Part 2 of 2.)

2.7 Images (cont.)

- The `img` element's `src` attribute specifies an image's location
- Every `img` element must have an `alt` attribute, which contains text that is displayed if the client cannot render the image
 - The `alt` attribute makes web pages more accessible to users with disabilities, especially vision impairments
 - `width` and `height` are optional attributes
 - If omitted, the browser uses the image's actual width and height
 - Images are measured in pixels

2.7 Images (Cont.)

alt Attribute

- A browser may not be able to render an image.
- Every `img` element in an HTML5 document must have an alt attribute.
- If a browser cannot render an image, the browser displays the alt attribute's value.
- The alt attribute is also important for accessibility—speech synthesizer software can speak the alt attribute's value so that a visually impaired user can understand what the browser is displaying. For this reason, the alt attribute should describe the image's contents.

2.7 Images (Cont.)

Using Images as Hyperlinks

- By using images as hyperlinks, you can create graphical web pages that link to other resources.
- In Fig. 2.7, we create five different image hyperlinks.
- Clicking an image in this example takes the user to a corresponding web page—one of the other examples in this chapter.

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.7: nav.html -->
4  <!-- Images as link anchors. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Navigation Bar</title>
9      </head>
10
11     <body>
12         <p>
13             <a href = "links.html">
14                 <img src = "buttons/links.jpg" width = "65"
15                     height = "50" alt = "Links">
16             </a>
17
18             <a href = "list.html">
19                 <img src = "buttons/list.jpg" width = "65"
20                     height = "50" alt = "List of Features">
21             </a>
22
```

Fig. 2.7 | Images as link anchors. (Part 1 of 3.)

EXAMPLE (cont..)

```
23      <a href = "contact.html">
24          <img src = "buttons/contact.jpg" width = "65"
25              height = "50" alt = "Contact Me">
26      </a>
27
28      <a href = "table1.html">
29          <img src = "buttons/table.jpg" width = "65"
30              height = "50" alt = "Tables Page">
31      </a>
32
33      <a href = "form.html">
34          <img src = "buttons/form.jpg" width = "65"
35              height = "50" alt = "Feedback Form">
36      </a>
37      </p>
38  </body>
39 </html>
```

Fig. 2.7 | Images as link anchors. (Part 2 of 3.)

EXAMPLE- Output



Fig. 2.7 | Images as link anchors. (Part 3 of 3.)

2.8 Special Characters and Horizontal Rules

- HTML5 provides **character entity references** (in the form `&code;`) for representing special characters that cannot be rendered otherwise
- The code can be:
 - Word abbreviations
 - Numbers
 - Decimal
 - Hexadecimal

EXAMPLE

| Symbol | Description | Character entity reference |
|---------------------------------|--------------|----------------------------|
| HTML5 character entities | | |
| & | ampersand | & |
| , | apostrophe | &apos |
| > | greater-than | > |
| < | less-than | < |
| " | quote | " |
| Other common character entities | | |
| non-breaking space | | &nbsp |
| © | copyright | © |
| — | em dash | — |
| – | en dash | – |
| ¼ | fraction 1/4 | ¼ |
| ½ | fraction 1/2 | ½ |

Fig. 2.8 | Some common HTML character entity references.

EXAMPLE

| Symbol | Description | Character entity reference |
|---------------|----------------------|----------------------------|
| $\frac{3}{4}$ | fraction 3/4 | ¾ |
| ... | horizontal ellipsis | … |
| ® | registered trademark | ® |
| § | section | § |
| ™ | trademark | ™ |

Fig. 2.8 | Some common HTML character entity references.

2.8 Special Characters and Horizontal Rules (Cont.)

- Figure 2.9 demonstrates how to use special characters in an HTML5 document.
- For an extensive list of character entities, see
www.w3.org/TR/REC-html40/sgml/entities.html

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.9: contact2.html -->
4  <!-- Inserting special characters. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Contact Page</title>
9      </head>
10
11     <body>
12         <p>
13             <a href = "mailto:deitel@deitel.com">Send an email to
14                 Deitel &amp; Associates, Inc.</a>.
15         </p>
16
17         <hr> <!-- inserts a horizontal rule -->
18
19         <!-- special characters are entered -->
20         <!-- using the form &code; -->
21         <p>All information on this site is <strong>&copy;
22             Deitel & Associates, Inc. 2012.</strong> </p>
23
24         <!-- to strike through text use <del> element -->
```

Fig. 2.9 | Inserting special characters. (Part 1 of 2.)

EXAMPLE

```
25      <!-- to subscript text use <sub> element -->
26      <!-- to superscript text use <sup> element -->
27      <!-- these elements are nested inside other elements -->
28      <p><del>You may download  $3.14 \times 10^2$ </sup>
29          characters worth of information from this site.</del>
30          The first item in the series is  $x_1$ .</p>
31          <p>Note: <math>\frac{1}{4}</math> of the information
32              presented here is updated daily.</p>
33      </body>
34  </html>
```

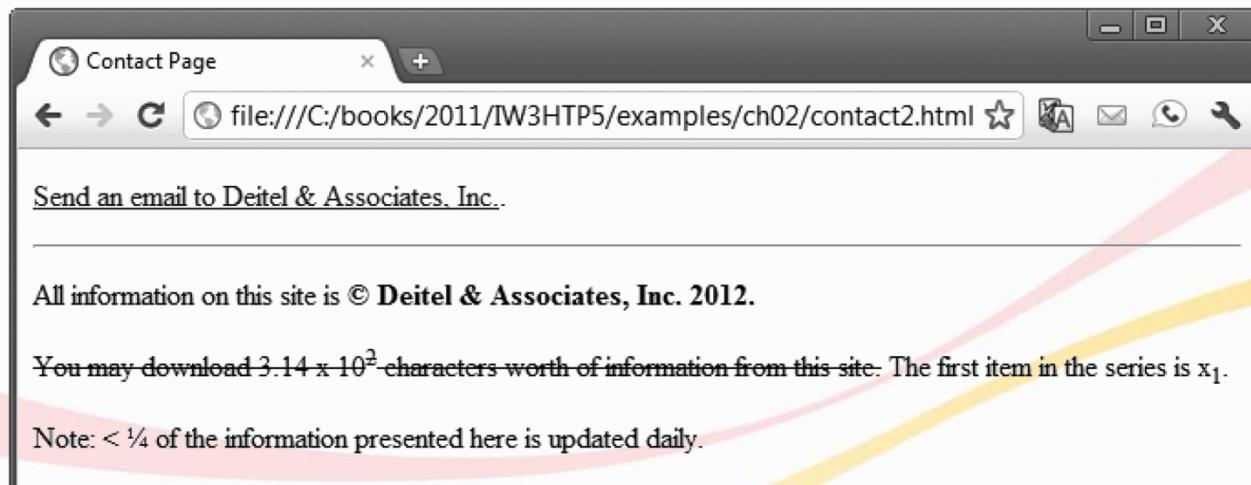


Fig. 2.9 | Inserting special characters. (Part 2 of 2.)

2.8 Special Characters and Horizontal Rules (Cont.)

- A **horizontal rule**, indicated by the `<hr>` tag renders a horizontal line with extra space above and below it in most browsers.
- The horizontal rule element should be considered a legacy element and you should avoid using it.
- CSS can be used to add horizontal rules and other formatting to documents.
- Special characters can also be represented as **numeric character references**—decimal or hexadecimal (hex) values representing special characters.
 - For example, the & character is represented in decimal and hexadecimal notation as `&` and `&x26;`, respectively.
- Hexadecimal numbers are discussed in Appendix E, Number Systems, which is available online at www.deitel.com/books/iw3htp5/.

2.9 Lists

- Unordered list element `ul`
 - creates a list in which each item in the list begins with a bullet symbol (typically a disc)
 - Each entry is an `li` (list item) element. Most web browsers render these elements with a line break and a bullet symbol at the beginning of the line.

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.10: links2.html -->
4  <!-- Unordered list containing hyperlinks. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Links</title>
9      </head>
10
11     <body>
12         <h1>Here are my favorite sites</h1>
13         <p><strong>Click on a name to go to that page</strong></p>
14
15         <!-- create an unordered list -->
16         <ul>
17             <!-- the list contains four list items -->
18             <li><a href = "http://www.youtube.com">YouTube</a></li>
19             <li><a href = "http://www.wikipedia.org">Wikipedia</a></li>
20             <li><a href = "http://www.amazon.com">Amazon</a></li>
21             <li><a href = "http://www.linkedin.com">LinkedIn</a></li>
22         </ul>
23     </body>
24 </html>
```

Fig. 2.10 | Unordered list containing hyperlinks. (Part I of 2.)

EXAMPLE

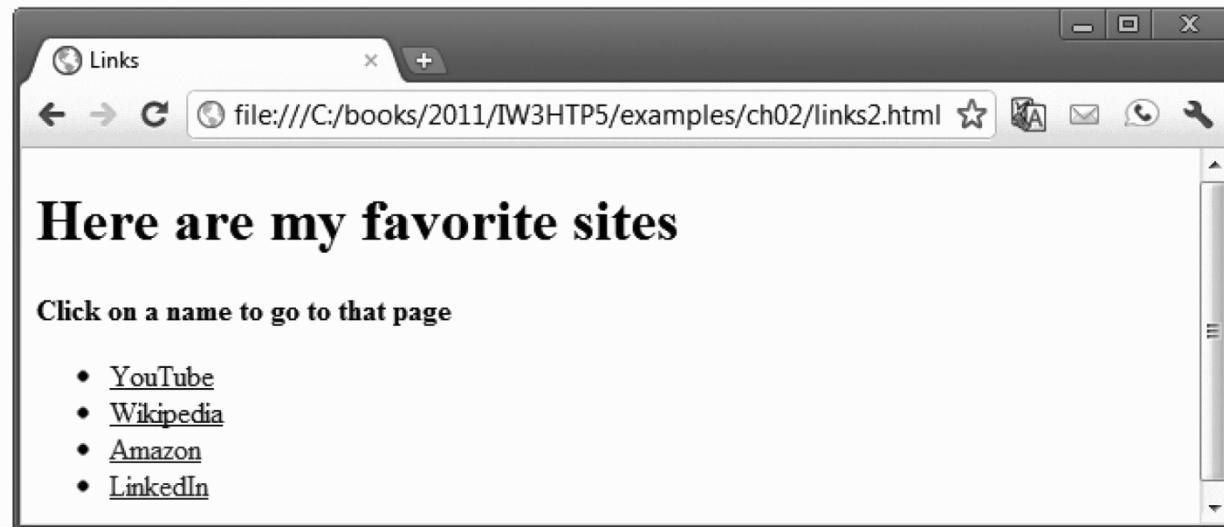


Fig. 2.10 | Unordered list containing hyperlinks. (Part 2 of 2.)

2.9 Lists (Cont.)

Nested Lists

- Lists may be *nested* to represent *hierarchical* relationships, as in a multi-level outline.
- Figure 2.11 demonstrates nested lists and ordered lists.
- The ordered-list element **ol** creates a list in which each item begins with a number.

EXAMPLE

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.11: list.html -->
4  <!-- Nested lists and ordered lists. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Lists</title>
9      </head>
10
11     <body>
12         <h1>The Best Features of the Internet</h1>
13
```

Fig. 2.11 | Nested lists and ordered lists. (Part 1 of 4.)

EXAMPLE

```
14      <!-- create an unordered list -->
15      <ul>
16          <li>You can meet new people from countries around
17              the world.</li>
18          <li>
19              You have access to new media as it becomes public:
20
21          <!-- this starts a nested unordered list, which uses a -->
22          <!-- different bullet. The list ends when you -->
23          <!-- close the <ul> tag. -->
24          <ul>
25              <li>New games</li>
26              <li>New applications
27
28          <!-- nested ordered list -->
29          <ol>
30              <li>For business</li>
31              <li>For pleasure</li>
32          </ol>
33          </li> <!-- ends line 27 new applications li-->
34
35          <li>Around the clock news</li>
36          <li>Search engines</li>
37          <li>Shopping</li>
```

Fig. 2.11 | Nested lists and ordered lists. (Part 2 of 4.)

EXAMPLE

```
38      <li>Programming
39
40          <!-- another nested ordered list -->
41          <ol>
42              <li>XML</li>
43              <li>Java</li>
44              <li>HTML5</li>
45              <li>JavaScript</li>
46              <li>New languages</li>
47          </ol>
48          </li> <!-- ends programming li of line 38 -->
49          </ul> <!-- ends the nested list of line 24 -->
50      </li>
51
52      <li>Links</li>
53      <li>Keeping in touch with old friends</li>
54          <li>It's the technology of the future!</li>
55      </ul> <!-- ends the unordered list of line 15 -->
56  </body>
57 </html>
```

Fig. 2.11 | Nested lists and ordered lists. (Part 3 of 4.)

EXAMPLE- Output

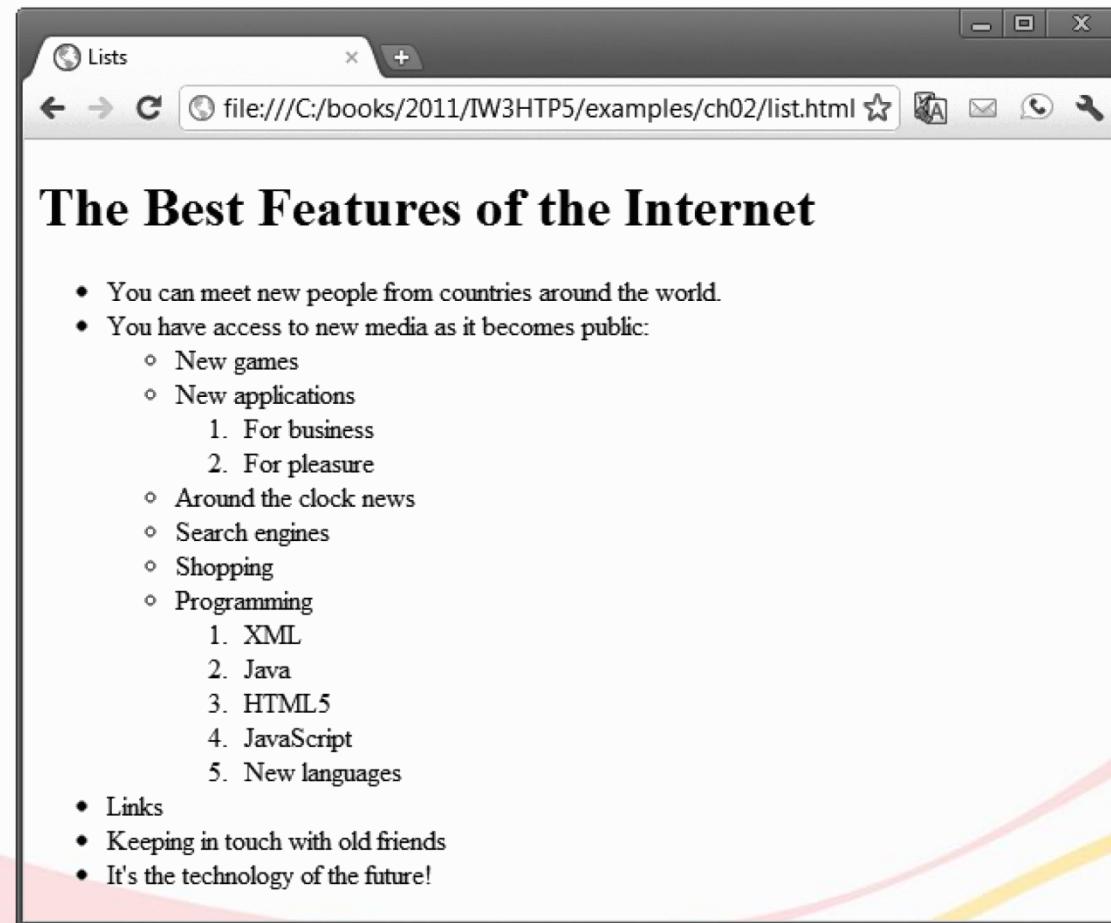


Fig. 2.11 | Nested lists and ordered lists. (Part 4 of 4.)

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.)

The screenshot shows a web browser window with the title "A simple HTML5 table". The page content is a table with the following structure:

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

Annotations with arrows point to specific parts of the table:

- Table caption: Points to the header text "Table of Fruits (1st column) and Their Prices (2nd column)".
- Table header: Points to the first row of the table, which contains column headers "Fruit" and "Price".
- Table body: Points to the rows containing the fruit names and their prices.
- Table footer: Points to the last row of the table, which contains the total values.
- Table border: Points to the outer border of the table.

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 1 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 1 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 with a dark gray header bar. In the header, there is a globe icon followed by the word "Forms". Below the header is a toolbar with standard navigation icons (back, forward, search, etc.) and a URL bar displaying "file:///C:/books/2011/IW3HTP5/examples/ch02/form.html". The main content area has a title "Feedback Form" in bold black font. Below the title is a message "Please fill out this form to help us improve our site." A text input field labeled "Name:" is present, followed by two buttons: "Submit" and "Clear". The browser window is set against a background with abstract pink and yellow curved shapes.

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
```

Difference between GET and POST method in HTTP Protocol

- 1) GET method passes request parameter in URL String while POST method passes request parameter in request body.
- 2) GET request can only pass limited amount of data while POST method can pass large amount of data to server.
- 3) GET request can be bookmarked and cached unlike POST requests.
- 4) GET is mostly used for view purpose (e.g. SQL SELECT) while POST is mainly use for update purpose (e.g. SQL INSERT or UPDATE).

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
61      </p>
62
63      <!-- <input type = "radio"> creates a radio -->
64      <!-- button. The difference between radio buttons -->
65      <!-- and checkboxes is that only one radio button -->
66      <!-- in a group can be selected. -->
67      <p>
68          <strong>How did you get to our site?</strong><br>
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" containing a "Feedback Form". The form includes the following components:

- Title:** Feedback Form
- Text:** Please fill out this form to help us improve our site.
- Name:**
- Comments:**
Enter comments here.
- E-mail Address:**
- Things you liked:**
Site design Links Ease of use Images Source code
- How did you get to our site?:**
Search engine Links from another site Deitel.com Web site Reference in a book Other
- Rate our site:**
Amazing 10
9
8
7
6
5
4
3
2
1
Awful

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.

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

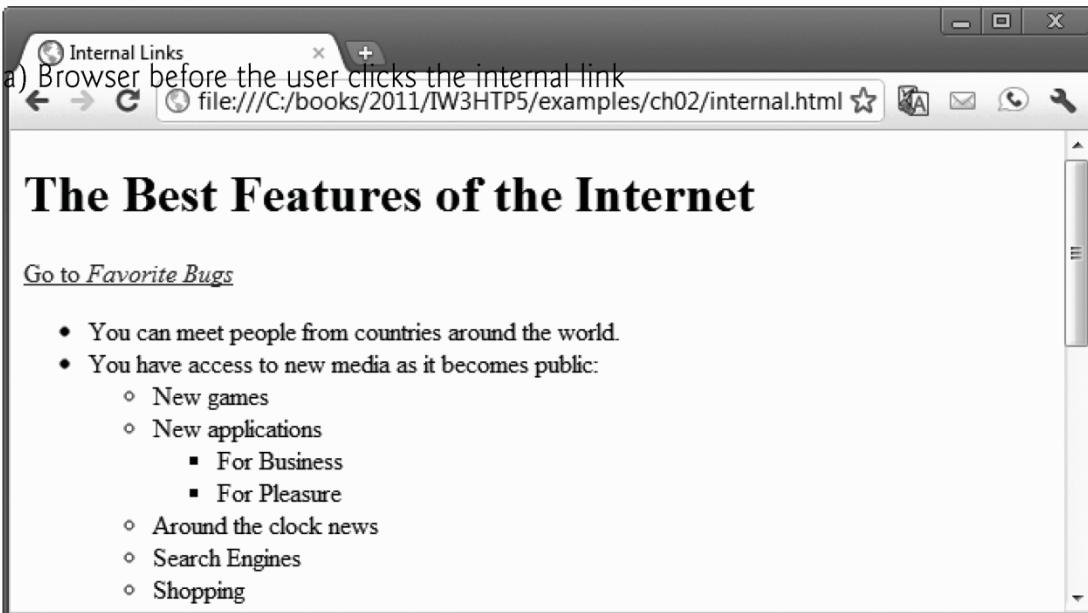


Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 4 of 5.)

EXAMPLE-Output

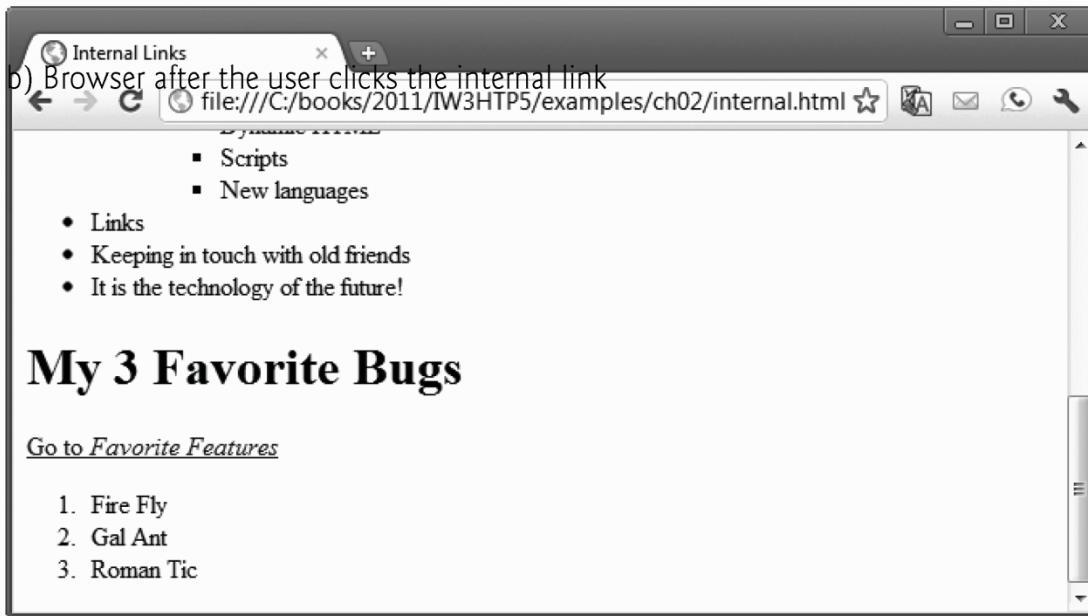


Fig. 2.16 | Internal hyperlinks to make pages more navigable. (Part 5 of 5.)

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...)

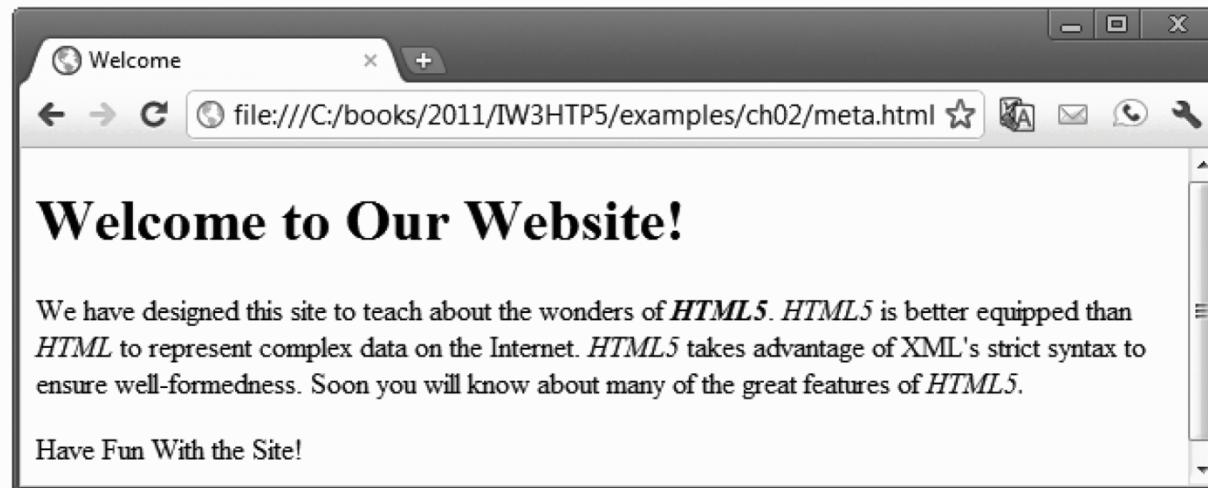


Fig. 2.17 | meta elements provide keywords and a description of a page. (Part 3 of 3.)

More element in HTML5

| Tag | Description |
|---------------------------------|---|
| <code><article></code> | Defines an article in a document |
| <code><aside></code> | Defines content aside from the page content |
| <code><bdi></code> | Isolates a part of text that might be formatted in a different direction from other text outside it |
| <code><details></code> | Defines additional details that the user can view or hide |
| <code><dialog></code> | Defines a dialog box or window |
| <code><figcaption></code> | Defines a caption for a <code><figure></code> element |
| <code><figure></code> | Defines self-contained content |
| <code><footer></code> | Defines a footer for a document or section |
| <code><header></code> | Defines a header for a document or section |
| <code><main></code> | Defines the main content of a document |
| <code><mark></code> | Defines marked/highlighted text |
| <code><menuitem></code> | Defines a command/menu item that the user can invoke from a popup menu |
| <code><meter></code> | Defines a scalar measurement within a known range (a gauge) |
| <code><nav></code> | Defines navigation links |
| <code><progress></code> | Represents the progress of a task |
| <code><rp></code> | Defines what to show in browsers that do not support ruby annotations |
| <code><rt></code> | Defines an explanation/pronunciation of characters (for East Asian typography) |
| <code><ruby></code> | Defines a ruby annotation (for East Asian typography) |
| <code><section></code> | Defines a section in a document |
| <code><summary></code> | Defines a visible heading for a <code><details></code> element |
| <code><time></code> | Defines a date/time |
| <code><wbr></code> | Defines a possible line-break |