HTML BASICS

- HEADINGS
- LINKING
- IMAGES
- USING IMAGES AS HYPERLINKS
- SPECIAL CHARACTERS AND HORIZONTAL RULES
- LISTS
- TABLES
- FORMS
- INTERNAL LINKING

Introduction

- HTML is a **markup language** that specifies the *structure* and *content* of documents that are displayed in web browsers.

First Example

```
1.  <!DOCTYPE html>

2.  <!-- First HTML example. -->
3.  <html>
4.      <head>
5.          <title>Welcome</title>
6.      </head>

7.      <body>
8.          <p>Welcome to HTML!</p>
9.      </body>
10. </html>
```

«DOCTYPE»

- The **document type declaration** (**DOCTYPE**) in line 1 is *required* in HTML documents so that browsers render the page in **standards mode**, according to the HTML and CSS specifications.
- You'll include the DOCTYPE in each HTML5 document you create.

Blank Lines

- We include blank lines (lines 2 and 8) to make our documents easier to read—the browser ignores them.

Comments

- Lines 3 is a **HTML comment**.
- You'll insert comments in your HTML markup to improve readability and describe the content of a document.
- The browser ignores comments when your document is rendered.
- HTML comments start with **<!--** and end with **-->**.
- We include in our examples comments that specify the figure number and file name and state the example's purpose. We'll often include additional comments, especially to explain new features.

Html, Head and Body Elements

- **HTML 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.
- Some important elements are the **html element** (which starts in line 3 and ends in line 10), the **head element** (lines 4–6) and the **body element** (lines 7–9).

Html, Head and Body Elements

- 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 HTML document. (document title, character set, styles, links, scripts, and other meta information)
- The **body section** contains the page's content, which the browser displays when the user visits the web page.

Start & End Tags

- HTML documents delimit most elements with a start tag and an 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 (</html>).

«title» Element

- 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.
- Search engines use the title for indexing purposes and when displaying results.

Paragraph Element (<P>…</P>)

- Some elements, such as the paragraph element denoted with and , help define the structure of a document.
- All the text placed between the and tags forms one paragraph. When a browser renders a paragraph, it places extra space above and below the paragraph text.

Headings

```
1.  <!DOCTYPE html>
2.  <!-- Heading elements h1 through h6. -->
3.  <html>
4.      <head>
5.          <title>Headings</title>
6.      </head>

7.      <body>
8.          <h1>Level 1 Heading</h1>
9.          <h2>Level 2 heading</h2>
10.         <h3>Level 3 heading</h3>
11.         <h4>Level 4 heading</h4>
12.         <h5>Level 5 heading</h5>
13.         <h6>Level 6 heading</h6>
14.     </body>
15. </html>
```

Headings


Headings
file:///C:/books/2011/IW3HTP5/examples/ch02/heading.html
Level 1 Heading
Level 2 heading
Level 3 heading
Level 4 heading
Level 5 heading
Level 6 heading

Headings

- Some text in an HTML document may be more important than other text. HTML provides six **heading elements** (h1 through h6) for specifying the *relative importance* of information.
- Heading element **h1** (line 8) is considered the ***most significant*** one and is typically rendered in a larger font than the other five. Each successive heading element (h2, h3, etc.) is typically rendered in a progressively ***smaller*** font.

Linking

- One of the most important HTML features is the **hyperlink**, which references (or links to) other resources, such as HTML documents and images.
- When a user clicks a **hyperlink**, the browser tries to execute an action associated with it (for example, navigate to a URL or open an e-mail client).
- Any displayed element can act as a **hyperlink**. Web browsers typically underline text hyperlinks and color their text blue by default so that users can distinguish hyperlinks from plain text.

Linking

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

Linking


Links
file:///C:/books/2011/IW3HTP5/examples/ch02/links.html
Here are my favorite sites:
Click a name to visit that site.
Facebook
Twitter
Foursquare
Google

Linking

- Line 10 introduces **strong element,** which indicates that its content has high importance. Browsers typically render such text in a bold font.
- Links are created using the **a** (**anchor**) element. Line 13 defines a ***hyperlink*** to the **URL** assigned to attribute **href** (hypertext reference), which specifies a resource's location, such as
  - a web page or location within a web page
  - a file
  - an e-mail address

Linking

- The anchor element in line 13 links the text Facebook to a web page located at **http://www.facebook.com**.
- The browser changes the color of any text link once you've clicked the link (in this case, the links are purple rather than blue).
- 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.

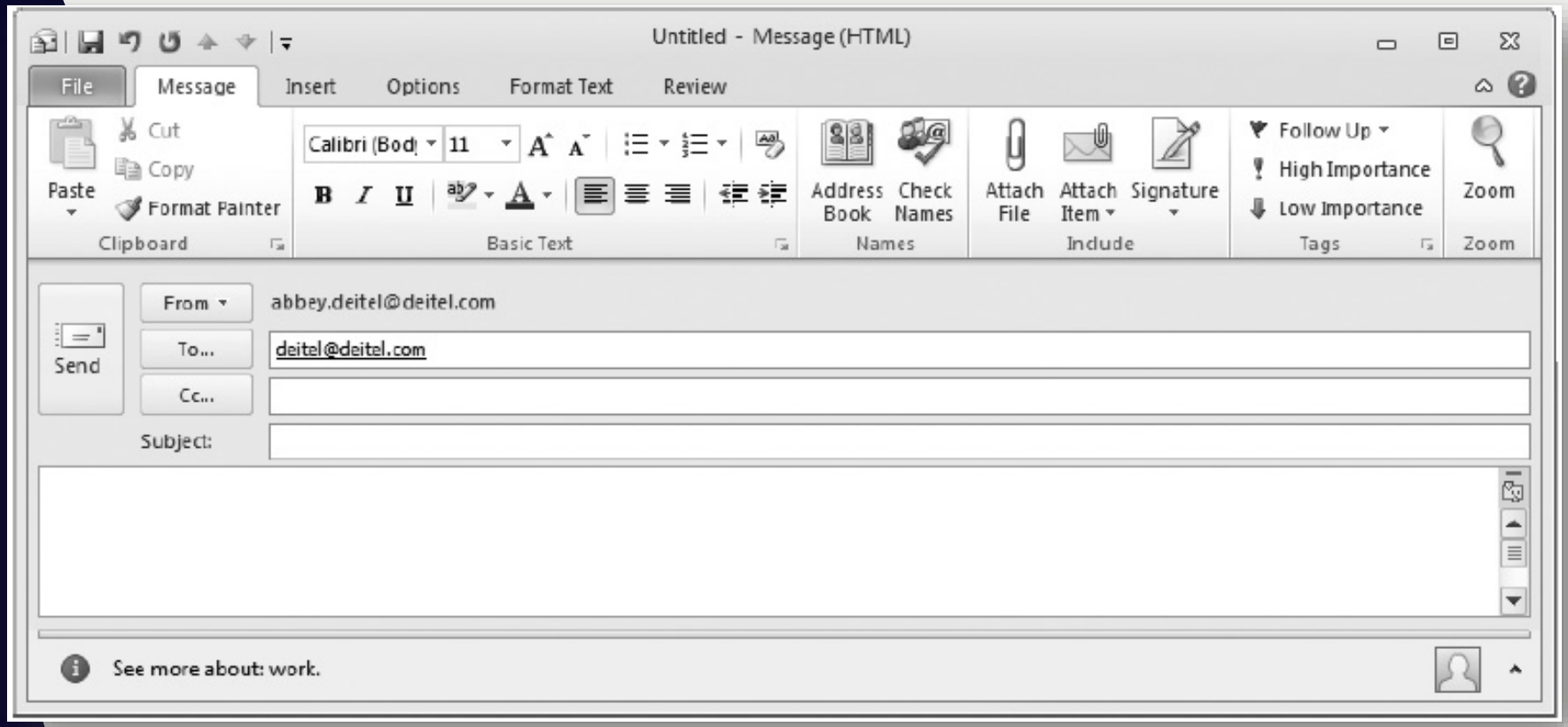
Hyperlinking to an e-mail address

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

Hyperlinking to an e-mail address

- Anchors can link to e-mail addresses using a **mailto:** URL. When the user clicks this type of anchored link, most browsers launch the user's default e-mail program (for example, Microsoft Outlook) to enable the user to write an email message to the linked address.
- Lines 13–14 contain an e-mail link. The form of an e-mail anchor is ….
- In this case, we link to the e-mail address deitel@deitel.com.
- Line 13 includes the e-mail address as it will appear in the message displayed on the browser.

Hyperlinking to an e-mail address



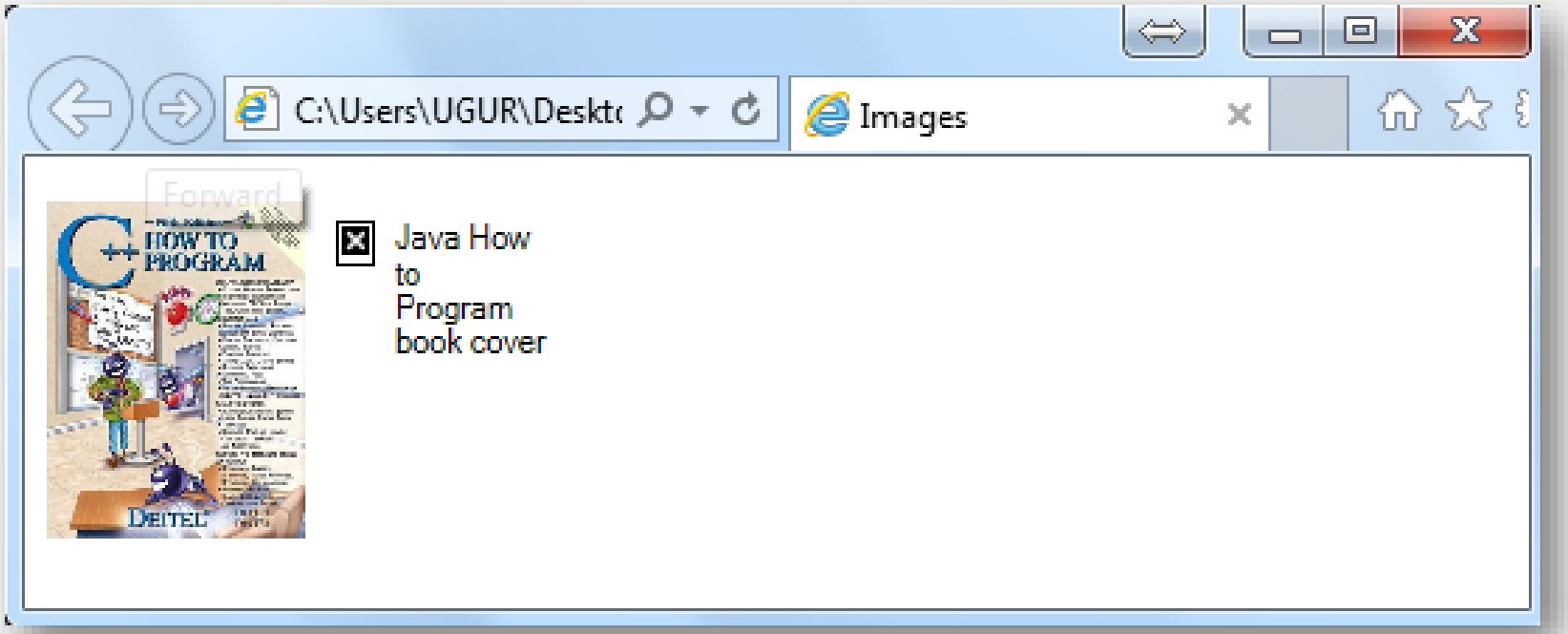
Images

- We've shown how to mark up documents that contain only text, but web pages may also contain images, animations, graphics, audios and even videos. The most popular *image formats* used by web developers today are *PNG (Portable Network Graphics)* and *JPEG (Joint Photographic Experts Group)*.

Images

```
1.  <!DOCTYPE html>
2.
3.  <!-- Including images in HTML files. -->
4.  <html>
5.      <head>
6.          <title>Images</title>
7.      </head>
8.
9.      <body>
10.         <p>
11.             <img src = "cpp.jpg" width = "92" height = "120"
12.                 alt = "C++ How to Program book cover">
13.             <img src = "java.jpg" width = "92" height = "120"
14.                 alt = "Java How to Program book cover">
15.         </p>
16.     </body>
17. </html>
```

Images


C:\Users\UGUR\Deskt
Images
Forward
C++ HOW TO PROGRAM
DEITEL
Java How to Program book cover

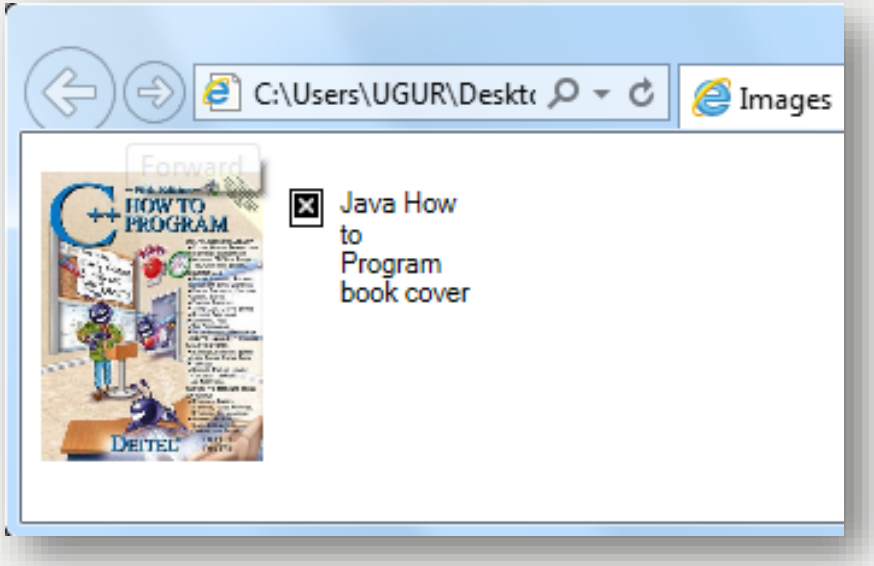
Images

- Lines 11–13 use an **img element** to include an image in the document. The image file's location is specified with the **src** (source) **attribute**. This image is located in the *same* directory as the HTML document, so only the image's file name is required.
- *Optional* attributes **width** and **height** specify the image's dimensions. You can *scale* an image by increasing or decreasing the values of the image width and height attributes. If these attributes are omitted, the browser uses the image's *actual* width and height.
- Images are measured in **pixels** ("picture elements"), which represent dots of color on the screen.

«alt» Attribute

- A browser may not be able to render an image for several reasons. It may not support images or the client may have disabled image viewing to reduce download time.
- Every img element in an HTML document *must* have an **alt attribute**. If a browser cannot render an image, the browser displays the alt attribute's value.
- The alt attribute should describe the image's contents.

Picture below shows the Internet Explorer browser rendering a red X symbol and displaying the alt attribute's value, signifying that the image (java.jpg) cannot be found.



Using Images as Hyperlinks

- By using images as hyperlinks, you can create graphical web pages that link to other resources.

```
1.  <!DOCTYPE html>
2.  <!-- Images as link anchors. -->
3.  <html>
4.    <head>
5.          <title>Navigation Bar</title>
6.    </head>
7.    <body>
8.          <p>
9.                <a href = "links.html">
10.               <img src = "buttons/links.jpg" width = "65" height = "50" alt = "Links">
11.               </a>

12.               <a href = "contact.html">
13.               <img src = "buttons/contact.jpg" width = "65" height = "50" alt = "Contact Me">
14.               </a>

15.               <a href = "form.html">
16.               <img src = "buttons/form.jpg" width = "65" height = "50" alt = "Feedback Form">
17.               </a>
18.         </p>
19.   </body>
20.  </html>
```

Using Images as Hyperlinks

- Lines 9–11 create an **image hyperlink** by nesting an img element in an anchor element.
- The **img** element's **src** attribute value specifies that this image (links.jpg) resides in a directory named buttons.
- The buttons directory and the HTML document are in the *same* directory.
- Images from other web documents also can be referenced by setting the src attribute to the name and location of the image. If you refer to an image on another website, the browser has to request the image resource from that site's server.
- Clicking an image hyperlink takes a user to the web page specified by the surrounding anchor element's href attribute.
- When the mouse *hovers* over a link of any kind, the URL that the link points to is displayed in the status bar at the bottom of the browser window.

Special Characters and Horizontal Rules

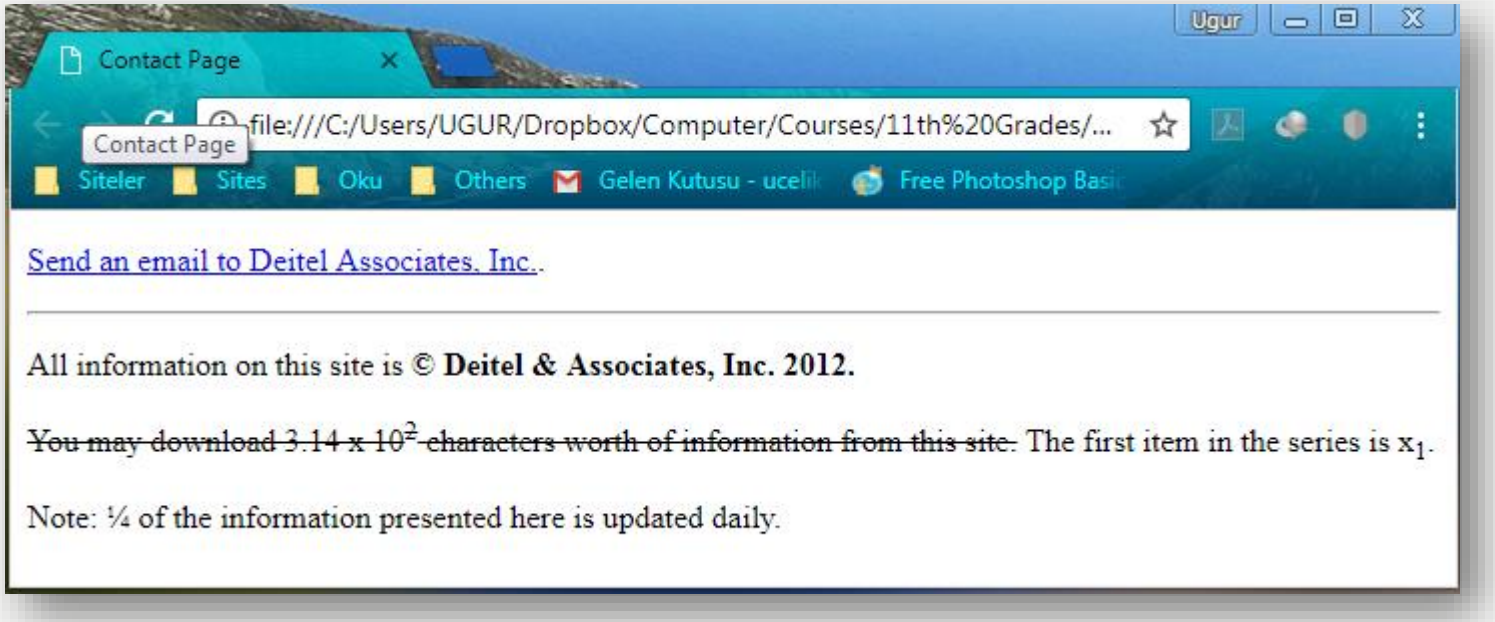
Symbol	Description	Character entity reference
&	ampersand	&amp;
’	apostrophe	&apos;
"	quote	&quot;
	non-breaking space	
©	copyright	&copy;
—	em dash	&mdash;
–	en dash	&ndash;
¼	fraction 1/4	&frac14;

HTML character entity references

Special Characters and Horizontal Rules

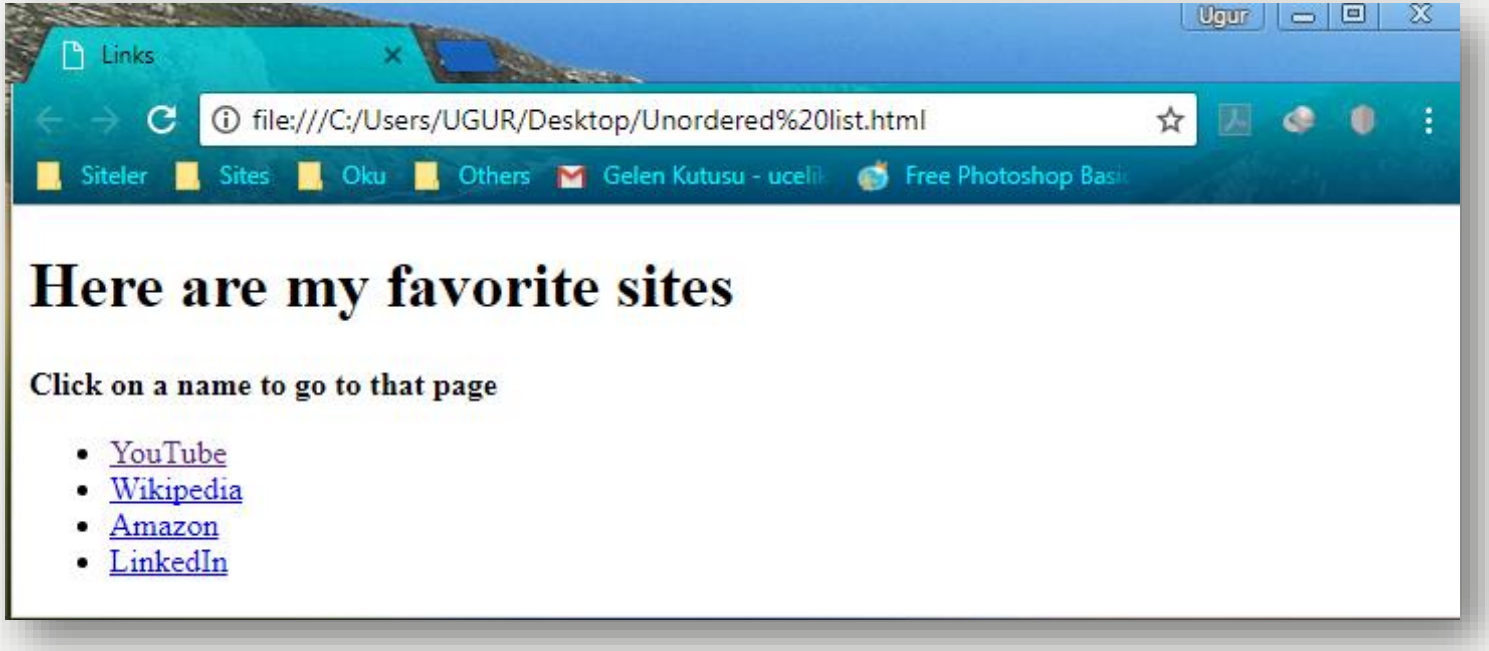
```
<!DOCTYPE html>
<html>
    <head>
        <title>Contact Page</title>
    </head>
    <body>
        <!-- special characters are entered -->
        <!-- using the form &code; -->
        <p>All information on this site is <strong>&copy;
        Deitel & Associates, Inc. 2012.</strong> </p>

        <hr> <!-- inserts a horizontal rule -->

        <!-- to strike through text use <del> element -->
        <!-- to subscript text use <sub> element -->
        <!-- to superscript text use <sup> element -->

        <p><del>You may download 3.14 x 10<sup>2</sup>
            characters worth of information from this site.</del>
            The first item in the series is x<sub>1</sub>.</p>
        <p>Note: &frac14; of the information
            presented here is updated daily.</p>
    </body>
</html>
```

Special Characters and Horizontal Rules



Lists

- An **unordered list** is a simple bullet-style list that does not order its items by letter or number.
- The unordered-list element **ul** creates a list in which each item begins with a bullet symbol (typically a *disc*). Each entry in an unordered list is an **li** (**list item**) element.

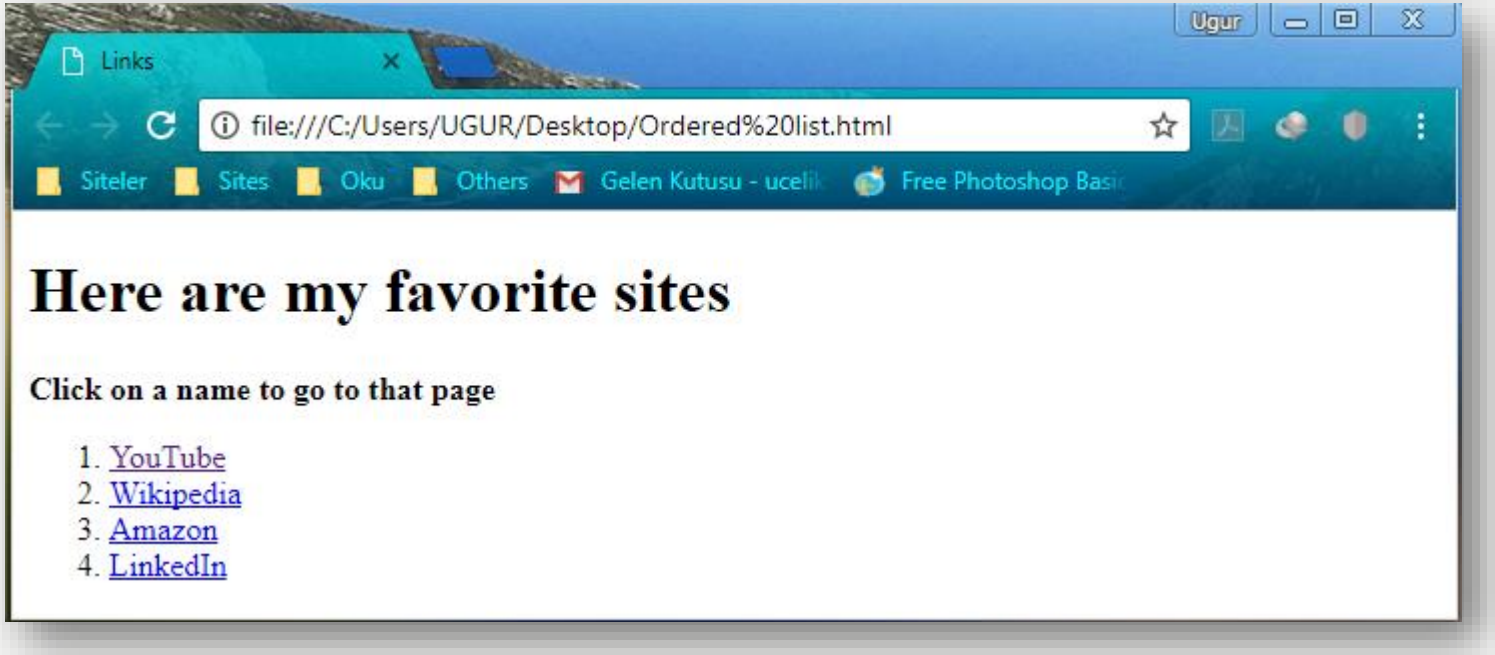


Lists

```
<!DOCTYPE html>
<html>
    <head>
        <title>Links</title>
    </head>
    <body>
        <h1>Here are my favorite sites</h1>
        <p><strong>Click on a name to go to that page</strong></p>
        <!-- create an unordered list -->
        <!-- the list contains four list items -->
        <ul>
            <li><a href = "http://www.youtube.com">YouTube</a></li>
            <li><a href = "http://www.wikipedia.org">Wikipedia</a></li>
            <li><a href = "http://www.amazon.com">Amazon</a></li>
            <li><a href = "http://www.linkedin.com">LinkedIn</a></li>
        </ul>
    </body>
</html>
```

Lists

- An **ordered list** element **(ol)** creates a list in which each item begins with a number.
- Each entry in an ordered list is an **li** (**list item**) element.



Lists

```
<!DOCTYPE html>
<html>
    <head>
        <title>Links</title>
    </head>
    <body>
        <h1>Here are my favorite sites</h1>
        <p><strong>Click on a name to go to that page</strong></p>
        <!-- create an ordered list -->
        <!-- the list contains four list items -->
        <ol>
            <li><a href = "http://www.youtube.com">YouTube</a></li>
            <li><a href = "http://www.wikipedia.org">Wikipedia</a></li>
            <li><a href = "http://www.amazon.com">Amazon</a></li>
            <li><a href = "http://www.linkedin.com">LinkedIn</a></li>
        </ol>
    </body>
</html>
```

Tables

- Tables are frequently used to organize data into *rows* and *columns*. Our first example creates a table with six rows and two columns to display price information for various fruits.
- Tables are defined with the **table** element. The **border** attribute with the value "1" specifies that the browser should place borders around the table and the table's cells.
- The **caption** element specifies a table's title. Text in this element is typically rendered above the table.

Tables

```
1.  <!DOCTYPE html>
2.  <html>
3.    <head>
4.          <title>A simple HTML5 table</title>
5.    </head>
6.
7.    <body>
8.    <!-- the <table> tag opens a table -->
9.    <table border = "1">
10.
11.         <!-- the <caption> tag summarizes the table's contents -->
12.         <caption><strong>Table of Fruits (1st column) and
13.               Their Prices (2nd column)</strong></caption>
14.
15.         <!-- the <thead> section formats the table header area -->
16.         <thead>
17.               <tr> <!-- inserts a table row -->
18.                     <th>Fruit</th> <!-- inserts a heading cell -->
19.                     <th>Price</th>
20.               </tr>
21.         </thead>
22.
```

Tables

```
23.         <!-- all table content is enclosed within the <tbody> -->
24.         <tbody>
25.             <tr>
26.                 <td>Apple</td>  <!-- insert a data cell -->
27.                 <td>$0.25</td>
28.             </tr>
29.             <tr>
30.                 <td>Orange</td>
31.                 <td>$0.50</td>
32.             </tr>
33.         </tbody>
34.         <!-- the <tfoot> section formats the table footer -->
35.         <tfoot>
36.             <tr>
37.                 <th>Total</th>
38.                 <th>$3.75</th>
39.             </tr>
40.         </tfoot>
41.
42.   </table>
43.   </body>
44. </html>
```

Tables

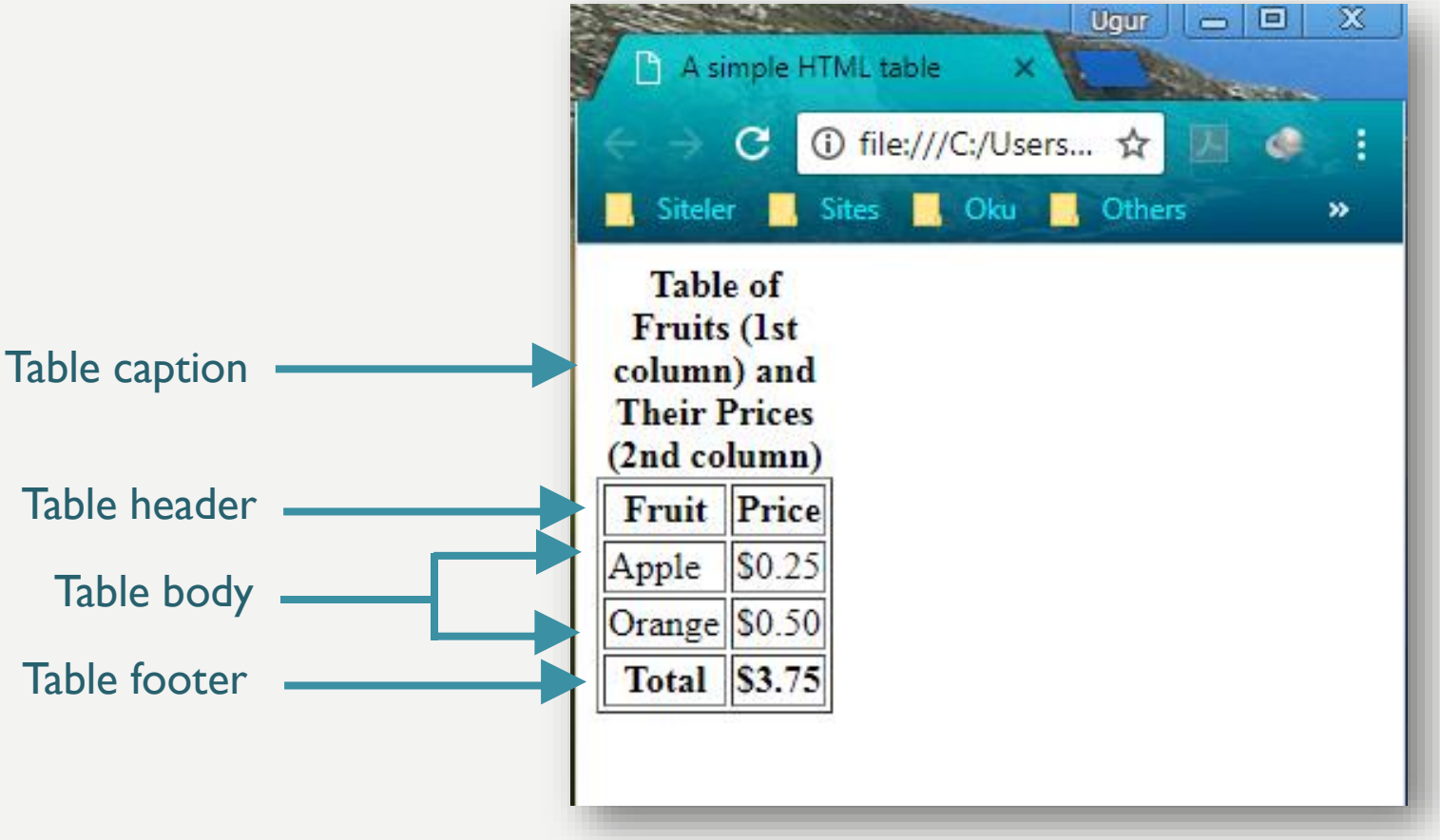


Table caption
Table header
Table body
Table footer
Ugur
A simple HTML table
file:///C:/Users...
Siteler
Sites
Oku
Others
Table of Fruits (1st column) and Their Prices (2nd column)
Fruit
Price
Apple
$0.25
Orange
$0.50
Total
$3.75

Tables

A table has three distinct sections—**head**, **body** and **foot**.

- The **head section** (or header cell) is defined with a **thead** element, which contains header information such as column names.
  - Each **tr** element defines an individual **table row**.
  - The **columns** in the thead section are defined with **th** elements. Most browsers **center** text formatted by **th** (table header column) elements and display them in **bold**. Table header elements are *nested* inside table row elements.

Tables

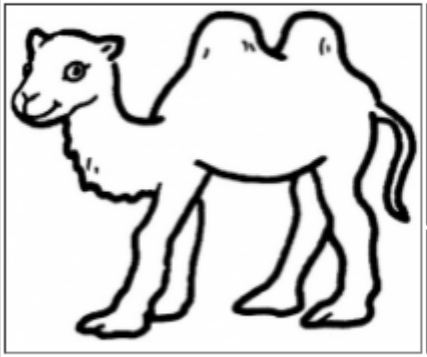
- The **body section**, or **table body**, contains the table's *primary data*. The table body is defined in a **tbody** element.
  - In the table body, each **tr** element specifies one row. **Data cells** contain individual pieces of data and are defined with **td** (**table data**) elements in each row.
- The **tfoot section** is defined with a **tfoot** (table foot) element. The text placed in the footer commonly includes *calculation results* and **footnotes**.
  - Here, we manually entered the calculation total.
  - Like other sections, the tfoot section can contain table rows, and each row can contain cells. As in the thead section, cells in the foot section are created using th elements, instead of the td elements used in the table body.

Using Rowspan & Colspan with Tables

- The values assigned to these attributes (**rowspan**, **colspan**)specify the number of rows or columns occupied by a cell.

Spanning Rows and Columns

A more complex sample table



Camelid comparison Approximate as of 10/2011				
	# of humps	Indigenous region	Spits?	Produces wool?
Camels (bactrian)	2	Africa/Asia	Yes	Yes
Llamas	1	Andes Mountains	Yes	Yes

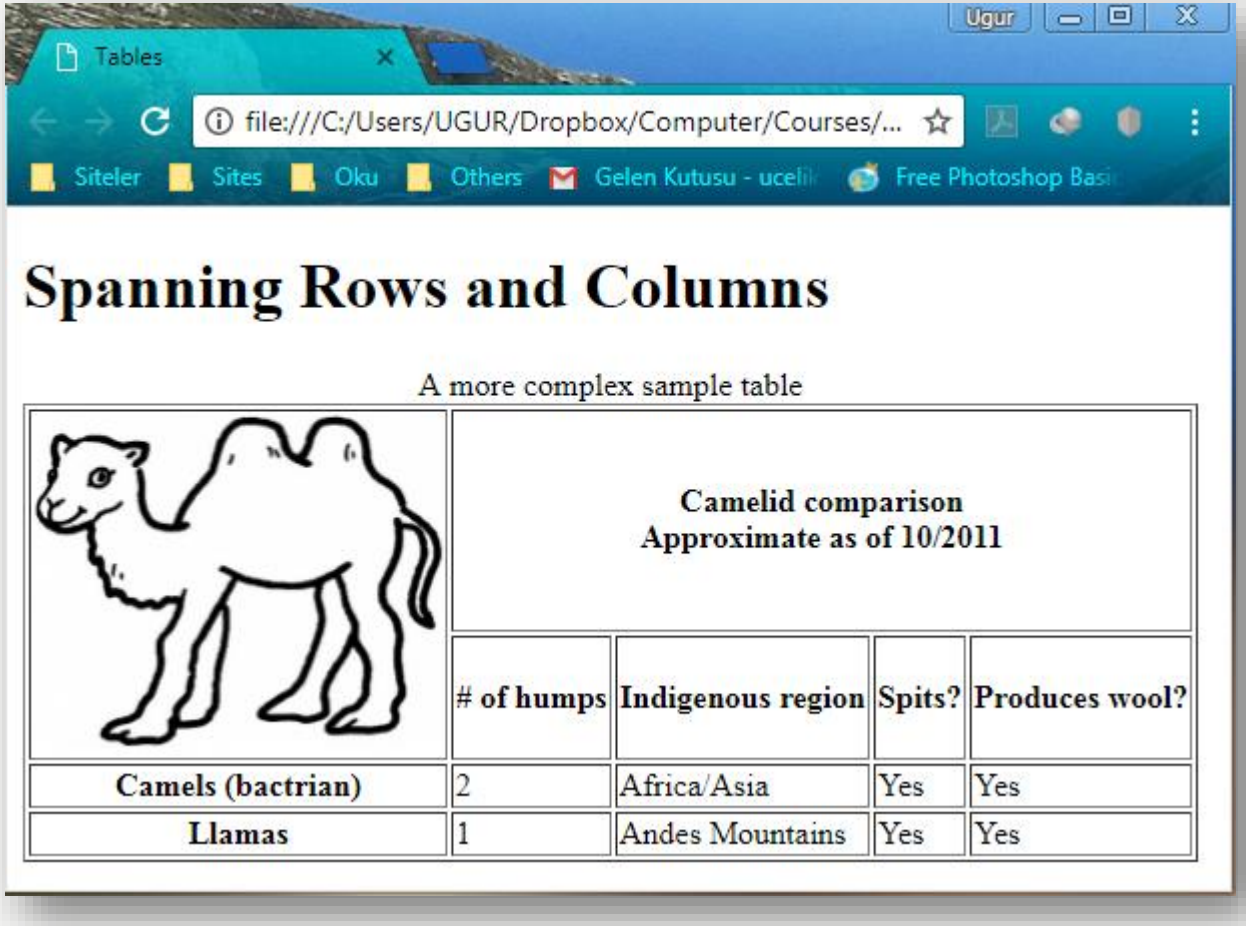
Using Rowspan & Colspan with Tables

```
1.  <!DOCTYPE html>
2.  <html>
3.    <head>
4.          <title>Tables</title>
5.    </head>
6.    <body>
7.          <h1>Spanning Rows and Columns</h1>
8.
9.          <table border = "1">
10.               <caption>A more complex sample table</caption>
11.               <thead>
12.                    <!-- rowspans and colspans merge the specified -->
13.                    <!-- number of cells vertically or horizontally -->
14.                    <tr>
15.                         <th rowspan = "2"> <!-- merge two rows -->
16.                              <img src = "camel.jpg" width = "205"
17.                              height = "167" alt = "Picture of a camel">
18.                         </th>
19.                         <th colspan = "4"> <!-- merge four columns -->
20.                              <strong>Camelid comparison</strong><br>
21.                              Approximate as of 10/2011
22.                         </th>
23.                    </tr>
```

Using Rowspan & Colspan with Tables

```
24.                     <tr>
25.                         <th># of humps</th>
26.                         <th>Indigenous region</th>
27.                         <th>Spits?</th>
28.                         <th>Produces wool?</th>
29.                     </tr>
30.                 </thead>
31.                 <tbody>
32.                     <tr>
33.                         <th>Camels (bactrian)</th>
34.                         <td>2</td>
35.                         <td>Africa/Asia</td>
36.                         <td>Yes</td>
37.                         <td>Yes</td>
38.                     </tr>
39.                     <tr>
40.                         <th>Llamas</th>
41.                         <td>1</td>
42.                         <td>Andes Mountains</td>
43.                         <td>Yes</td>
44.                         <td>Yes</td>
45.                     </tr>
46.                 </tbody>
47.             </table>
48.     </body>
49. </html>
```

Using Rowspan & Colspan with Tables



Spanning Rows and Columns

A more complex sample table

	Camelid comparison Approximate as of 10/2011			
	# of humps	**Indigenous region**	**Spits?**	**Produces wool?**
Camels (bactrian)	2	Africa/Asia	Yes	Yes
Llamas	1	Andes Mountains	Yes	Yes

Forms

- When browsing websites, users often need to provide information such as search queries, e-mail addresses and zip codes.
- HTML provides a mechanism, called a **form**, for collecting data from a user.
- Data that users enter on a web page is normally sent to a **web server** that provides access to a site's resources (for example, HTML documents, images, animations, videos).

Forms

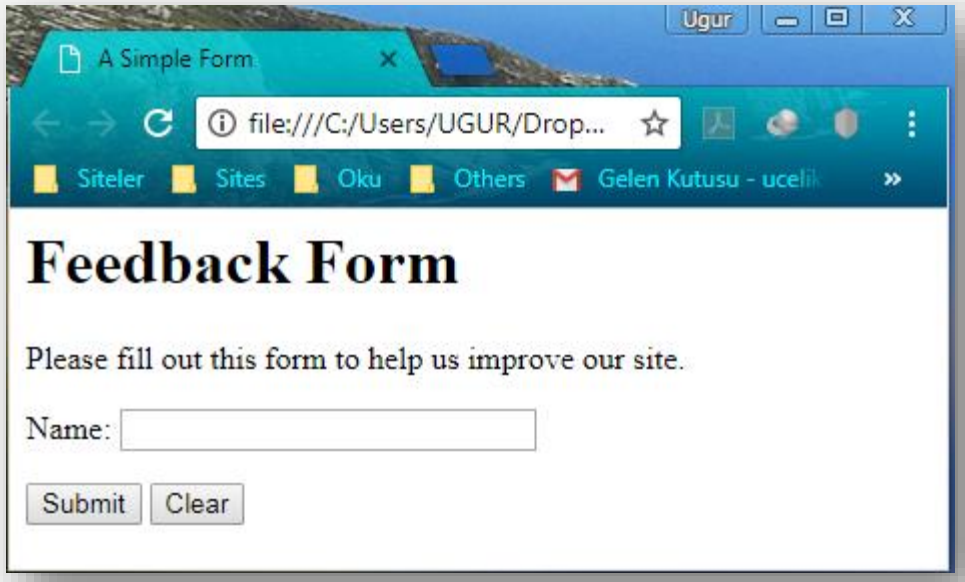
- These resources are located either on the same machine as the web server or on a machine that the web server can access through the Internet. When a browser requests a publicly available web page or file that's located on a server, the server processes the request and returns the requested resource. A request contains the *name* and *path* of the desired resource and the *protocol* (method of communication).
- HTML documents are requested and transferred via the **Hypertext Transfer Protocol** (**HTTP**).

Forms

```
1.  <html>
2.    <head>
3.    <title>A Simple Form</title>
4.    </head>
5.
6.    <body>
7.          <h1>Feedback Form</h1>
8.
9.          <p>Please fill out this form to help us improve our site.</p>
10.         <!-- this tag starts the form, gives the method of sending -->
11.         <!-- information and the location of form script -->
12.               <form method = "post" action = "">
13.               <p>
14.                     <!-- hidden inputs contain non-visual information -->
15.                     <input type = "hidden" name = "recipient"
16.                           value = "deitel@deitel.com" />
17.                     <input type = "hidden" name = "subject"
18.                           value = "Feedback Form" />
19.                     <input type = "hidden" name = "redirect"
20.                           value = "main.html" />
21.               </p>
```

Forms

```
22.             <!-- <input type = "text"> inserts a text box -->
23.             <p><label>Name:
24.                 <input name = "name" type = "text" size = "25"
25.                 maxlength = "30" />
26.             </label></p>
27.             <p>
28.                 <!-- input types "submit" and "reset" insert buttons for -->
29.                 <!-- submitting and clearing the form's contents -->
30.                 <input type = "submit" value = "Submit" />
31.                 <input type = "reset" value = "Clear" />
32.             </p>
33.         </form>
34.   </body>
35. </html>
```



Method attribute

- 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. This method of passing data to the server is *transparent*—the user doesn't see the data after the form is submitted.
  - 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.

Action attribute

- The **action** attribute in the form element specifies the URL of a *script on the web server* that will be invoked to process the form's data.
- 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 HTML document file names that act as links.

Text input

- The **text** input inserts a **text field** in the form. Users can type data in text fields. The label element provides users with information about the input element's purpose.
- The input element's **size** attribute specifies the number of characters visible in the text field. Optional attribute **maxlength** limits the number of characters input into the text field—in this case, the user is not permitted to type more than 30 characters.

Submit and Reset

- Two input elements create two buttons. 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.
- The value attribute of the reset input element sets the text displayed on the button (the default value is **Reset** if you omit the value attribute).

A Complex Form

```
1.  <html>
2.    <head>
3.      <title>A Complex Form</title>
4.    </head>
5.
6.    <body>
7.      <h1>A Complex Form</h1>
8.      <p>Please fill out this form to help
9.        us improve our site.</p>
10.
11.     <form method = "post" action = "">
12.       <p>
13.         <input type = "hidden" name = "recipient"
14.           value = "deitel@deitel.com" />
15.         <input type = "hidden" name = "subject"
16.           value = "Feedback Form" />
17.         <input type = "hidden" name = "redirect"
18.           value = "main.html" />
19.       </p>
```

A Complex Form

```
20.     <p><label>Name Surname:
21.         <input name = "name" type = "text" size = "25" />
22.         </label></p>
23.
24.     <p><label>E-mail Address:
25.         <input name = "email" type = "password"  size = "25" />
26.         </label></p>
27.
28.      <p><label>Comments:<br/>
29.        <textarea name = "comments"
30.          rows = "4" cols = "36">Enter comments here.</textarea>
31.      </label></p>
```

A Complex Form

```
32. <p>
33.         <strong>Things you liked:</strong><br/>
34.
35.         <label>Site design
36.           <input name = "thingsliked" type = "checkbox"
37.             value = "Design" /></label>
38.         <label>Links
39.           <input name = "thingsliked" type = "checkbox"
40.             value = "Links" /></label>
41.         <label>Ease of use
42.           <input name = "thingsliked" type = "checkbox"
43.             value = "Ease" /></label>
44.         <label>Images
45.           <input name = "thingsliked" type = "checkbox"
46.             value = "Images" /></label>
47.         <label>Source code
48.           <input name = "thingsliked" type = "checkbox"
49.             value = "Code" /></label>
50.       </p>
```

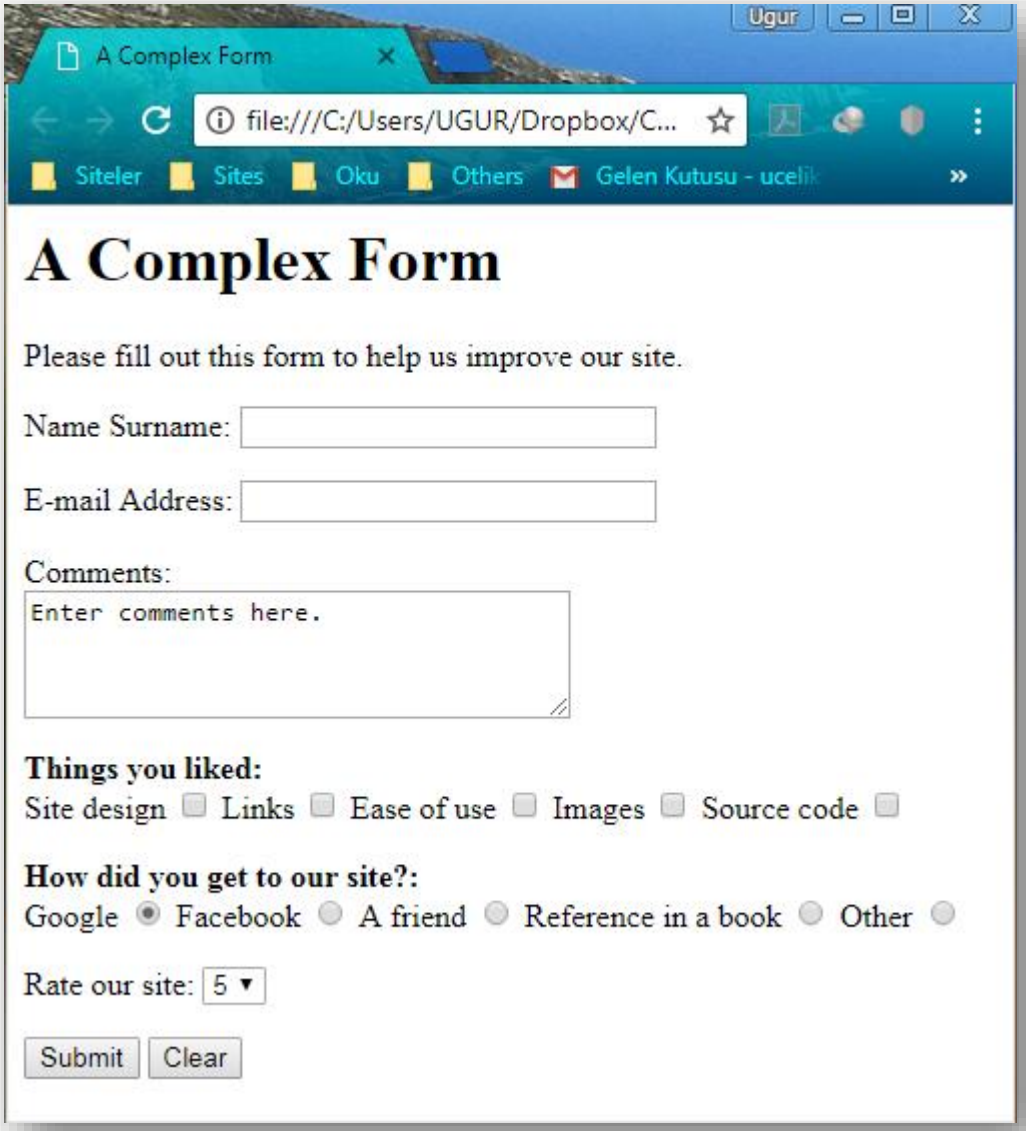
A Complex Form

```
51. <p>
52.         <strong>How did you get to our site?:</strong><br />
53.
54.         <label>Google
55.           <input name = "howtosite" type = "radio"
56.             value = "search engine" checked = "checked" /></label>
57.         <label>Facebook
58.           <input name = "howtosite" type = "radio"
59.             value = "link" /></label>
60.         <label>A friend
61.           <input name = "howtosite" type = "radio"
62.             value = "deitel.com" /></label>
63.         <label>Reference in a book
64.           <input name = "howtosite" type = "radio"
65.             value = "book" /></label>
66.         <label>Other
67.           <input name = "howtosite" type = "radio"
68.             value = "other" /></label>
69. </p>
```

A Complex Form

```
70.         <p>
71.           <label>Rate our site:
72.
73.             <select name = "rating">
74.               <option selected = "selected">5</option>
75.               <option>4</option>
76.               <option>3</option>
77.               <option>2</option>
78.               <option>1</option>
79.             </select>
80.           </label>
81.         </p>
82.
83.         <p>
84.           <input type = "submit" value = "Submit" />
85.           <input type = "reset" value = "Clear" />
86.         </p>
87.       </form>
88.     </body>
89.  </html>
```

A Complex Form



Textarea

- 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.
- In this example, the textarea is four rows high and 36 characters wide. To display *default text* in the textarea, place the text between the <textarea> and </textarea> tags. Default text can be specified in other input types, such as text fields, by using the value attribute.

Password Box

- The **password** input inserts a password box with the specified size (maximum number of displayed characters). A password box allows users to enter sensitive information, such as credit card numbers and passwords, by "masking" the information input with asterisks (*). The actual value input is sent to the web server, not the masking characters.

Checkbox

- Checkboxes enable users to select an option. When a user selects a checkbox, a *check mark* appears in the checkbox. Otherwise, the checkbox remains empty.
- Each checkbox input creates a new checkbox.
- Checkboxes can be used individually or in groups. checkboxes that belong to a group are assigned the same name (in this case, "thingsliked").

Radio button

- **Radio buttons** are similar to checkboxes, except that only one radio button in a group of radio buttons may be selected at any time. The radio buttons in a group all have the same name attributes and are distinguished by their different value attributes.

Select element

- The **select** element provides a *drop-down list* from which the user can select an item.
  - The name attribute identifies the drop-down list.
  - The **option** elements add items to the drop-down list. The option element's **selected** attribute specifies which item *initially* is displayed as the selected item in the select element.
  - If no option element is marked as selected, the browser selects the *first* option by default.

THE END

UGUR CELIK