KTU
NOTES

# Module 1

## Introduction to Computers and the Internet

**Internet:**A worldwide collection of networks that links millions of businesses, government agencies, educational institutions, and individuals.

### World Wide Web, HTML, HTTP

- The World Wide Web (abbreviated as WWW or W3, commonly known as the Web) is a system of interlinked hypertext documents that are accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.
- The World Wide Web allows computer users to execute web-based applications and to locate and view multimedia-based documents on almost any subject over the Internet.
- HTML: A special type of computer language called a markup language designed to specify the content and structure of web pages (also called documents) in a portable manner. HTML5, now under development, is the emerging version of HTML. HTML enables us to create content that will render appropriately across the extraordinary range of devices connected to the Internet—including smartphones, tablet computers, notebook computers, desktop computers, special-purpose devices such as large-screen displays at concert arenas and sports stadiums, and more.
- The URL (Uniform Resource Locator) specifies the address (i.e., location) of the web page displayed in the browser window. Each web page on the Internet is associated with a unique URL. URLs usually begin with http://.

### Web basics

- Fundamentals of web-based interactions between a client web browser and a web server.
- In its simplest form, a *web page* is nothing more than an HTML (HyperText Markup Language) document (with the extension .html or .htm) that describes to a web browser the document’s content and structure

### Hyperlinks

- HTML documents normally contain hyperlinks, which, when clicked, load a specified web document. Both images and text may be hyperlinked.
- When the mouse pointer hovers over a hyperlink, the default arrow pointer changes into a hand with the index finger pointing upward. Often hyperlinked text appears

underlined and in a different color from regular text in a web page.

- Hyperlinks are widely used to reference sources, or sites that have more information on a particular topic. The paths created by hyperlinking create the effect of the "web."
- When the user clicks a hyperlink, a web server locates the requested web page and sends it to the user's web browser.
- Similarly, the user can type the address of a web page into the browser's address field and press Enter to view the specified page.
- Hyperlinks can reference other web pages, e-mail addresses, files and more. If a hyperlink's URL is in the form mailto:emailAddress, clicking the link loads your default e-mail program and opens a message window addressed to the specified e-mail address. If a hyperlink references a file that the browser is incapable of displaying, the browser prepares to download the file, and generally prompts the user for information about how the file should be stored. When a file is downloaded, it's copied onto the user's computer. Programs, documents, images, sound and video files are all examples of downloadable files.

### URIs and URLs

- URIs (Uniform Resource Identifiers) identify resources on the Internet.
- URIs that start with http:// are called URLs (Uniform Resource Locators).
- Common URLs refer to files, directories or server-side code that performs tasks such as database lookups, Internet searches and business-application processing. If you know the URL of a publicly available resource anywhere on the web, you can enter that URL into a web browser's address field and the browser can access that resource.

## Introduction to HTML5

- Unlike programming languages, such as C, C++, C#, Java and Visual Basic, HTML5 is a markup language that specifies the *structure* and *content* of documents that are displayed in web browsers.

### Editing HTML5:

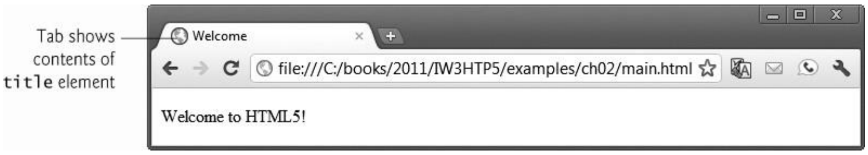
- We'll create HTML5 documents by typing HTML5 markup text in a text editor (such as Notepad, TextEdit, vi, emacs) and saving it with the .html or .htm filename extension.

- Computers called web servers store HTML5 documents. Clients (such as web browsers running on your local computer or smartphone) request specific resources such as HTML5 documents from web servers.

Example HTML5:

- This first example displays the message Welcome to HTML5! in the browser.

```
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>
```


Tab shows contents of title element
Welcome
file:///C:/books/2011/IW3HTP5/examples/ch02/main.html

Welcome to HTML5!

The above figure is an HTML5 document named main.html

- Document Type Declaration

The **document type declaration (DOCTYPE)** in line 1 is *required* in HTML5 documents so that browsers render the page in **standards mode**, according to the HTML and CSS specifications. Some browsers operate in **quirks mode** to maintain backward compatibility with web pages that are not up-to-date with the latest standards. We'll include the DOCTYPE in each HTML5 document we create.

- Blank Lines

We include blank lines (lines 2 and 10) to make our documents easier to read—the browser ignores them.

- Comments

Lines 3–4 are HTML5 comments. We'll 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. HTML5 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

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. Some important elements are:

- the html element (which starts in line 5 and ends in line 14)

- the head element (lines 6–9)

-the body element (lines 11–13).

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 uses (line 7)—which helps the browser determine how to render the content—and the title (line 8). 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..

- Start Tags and End Tags

HTML5 documents delimit most elements with a start tag and an end tag.
Start tag-consists of the element name in angle brackets (for example, <html> in line 5).
End tag-consists of the element name preceded by a forward slash (/) in angle brackets (for example, </html> in line 14). There are several so-called "void elements" that do not have end tags.

- Title element

-Line 8 specifies a title element. This 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.

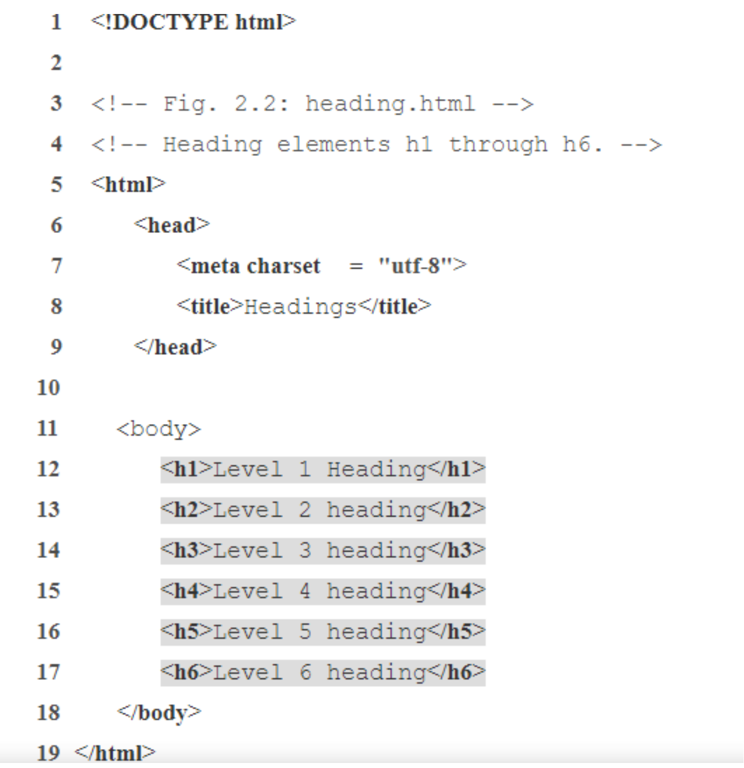
- Paragraph Element (<p>...</p>)

  - Some elements, such as the **paragraph element** denoted with <p> and </p> in line 12, help define the structure of a document.

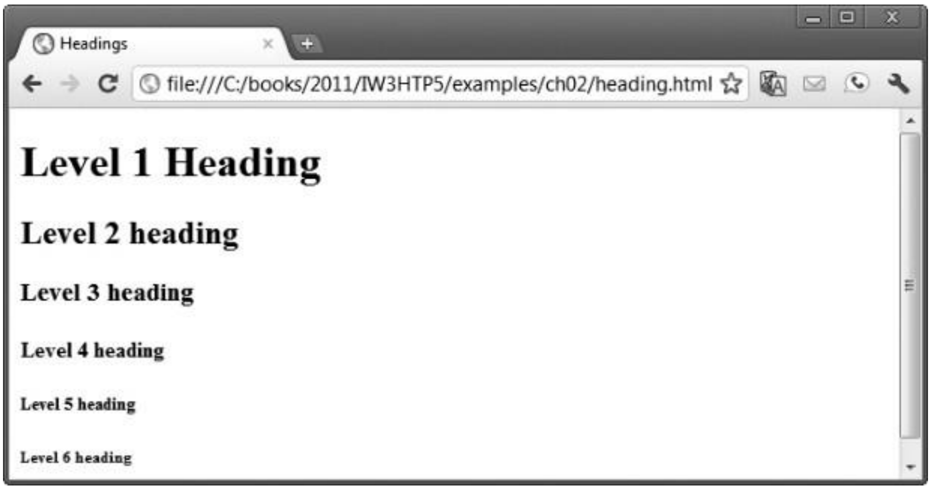
  - All the text placed between the <p> and </p> tags forms one paragraph.

## HEADINGS

- Some text in an HTML5 document may be more important than other text.
- HTML5 provides six heading elements (h1 through h6) for specifying the relative importance of information (shown in figure)
- Heading element h1 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.
- The text size used to display each heading element can vary between browsers.
- Placing a heading at the top of each page helps viewers understand the purpose of the page. Headers also help create an outline for a document and are indexed by search engines.



```
<!DOCTYPE html>

<!-- Fig. 2.2: heading.html -->
<!-- Heading elements h1 through h6. -->
<html>
    <head>
        <meta charset   =  "utf-8">
        <title>Headings</title>
    </head>

   <body>
       <h1>Level 1 Heading</h1>
       <h2>Level 2 heading</h2>
       <h3>Level 3 heading</h3>
       <h4>Level 4 heading</h4>
       <h5>Level 5 heading</h5>
       <h6>Level 6 heading</h6>
   </body>
</html>
```



Level 1 Heading

Level 2 heading

Level 3 heading

Level 4 heading

Level 5 heading

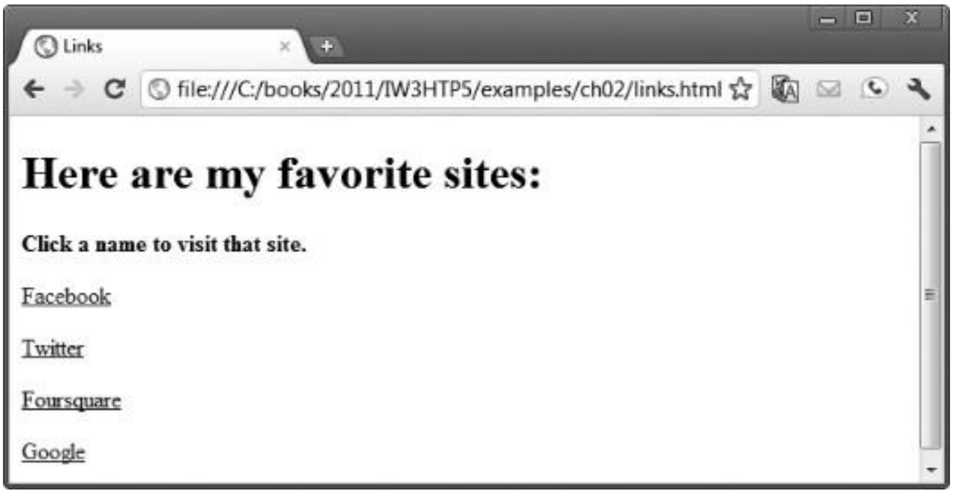
Level 6 heading

Figure:Heading elements h1 through h6

## LINKING

- One of the most important HTML5 features is the hyperlink, which references (or links to) other resources, such as HTML5 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. In Figure created text hyperlinks to four websites.

```
<!DOCTYPE html>

<!-- Fig. 2.3: links.html -->
<!-- Linking to other web pages. -->
<html>
    <head>
        <meta charset = "utf-8">
        <title>Links</title>
    </head>

    <body>
        <h1>Here are my favorite sites:</h1>
        <p><strong>Click a name to visit that site.</strong></p>

         <!-- create four text hyperlinks -->
         <p><a href = "http://www.facebook.com">Facebook</a></p>
         <p><a href = "http://www.twitter.com">Twitter</a></p>
         <p><a href = "http://www.foursquare.com">Foursquare</a></p>
         <p><a href = "http://www.google.com">Google</a></p>
     </body>
 </html>
```



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

Figure: Linking to other web pages.

- Line 13 introduces the 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 16 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
- The anchor element in line 16 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.

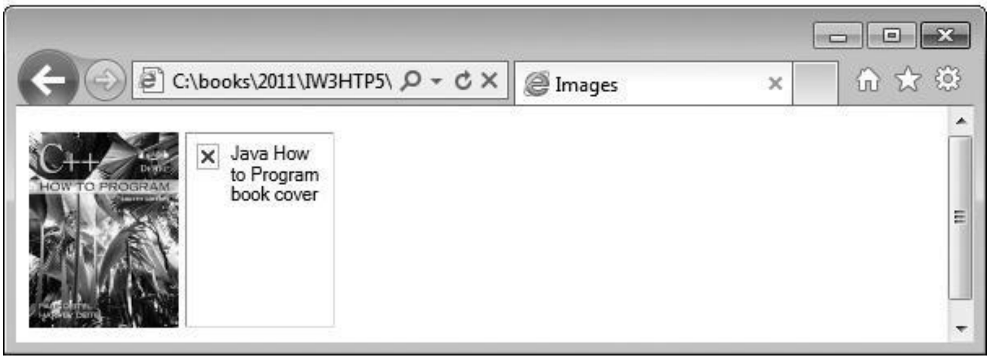
## IMAGES

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).

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. Figure below demonstrates how to include images in web pages.

Internet Explorer 9 showing an image and the `alt` text for a missing image



```
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 = "cpphtp.png" width = "92" height = "120"
14                  alt = "C++ How to Program book cover">
15              <img src = "jhtp.png" width = "92" height = "120"
16                  alt = "Java How to Program book cover">
17          </p>
18      </body>
19  </html>
```

- Lines 13–14 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 HTML5 document, so only the image's file name is required. This is known as a relative path—the image is stored relative to the

location of the document.

- 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. Image-editing programs display the dimensions, in pixels, of an image.

### - alt **Attribute**

- A browser may not be able to render an image for several reasons. It may not support images—as is the case with text-only browsers—or the client may have disabled image viewing to reduce download time. 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.

### -void Elements

- Some HTML5 elements (called void elements) contain only attributes and do not mark up text (i.e., text is not placed between a start and an end tag). Although this is not required in HTML5, we can terminate void elements (such as the img element) by using the forward slash character (/) inside the closing right angle bracket (>) of the start tag.

```
<img src = "jhtp.png" width = "92" height = "120" alt
= "Java How to Program book cover" />
```

## SPECIAL CHARACTERS AND HORIZONTAL RULES

- When marking up text, certain characters or symbols may be difficult to embed directly into an HTML5 document.
- Some keyboards do not provide these symbols (such as ©), or their presence in the markup may cause syntax errors (as with <). For example, the markup

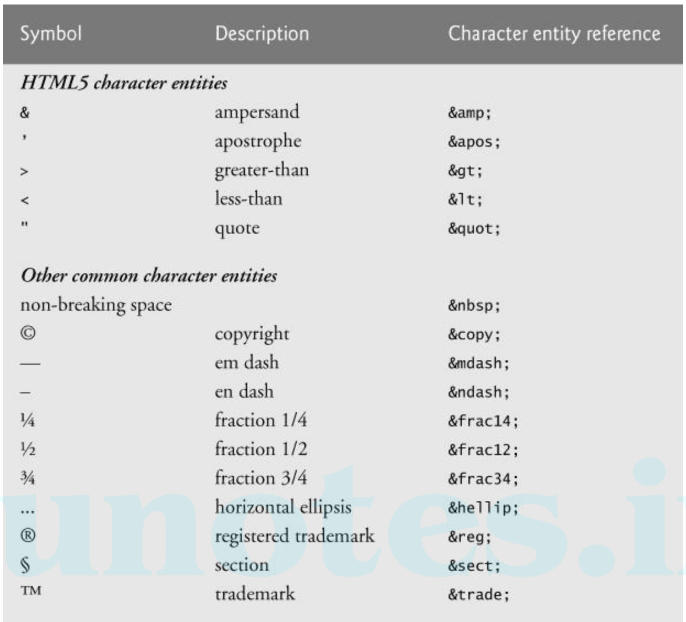
  **<p>if x < 10 then increment x by 1</p>**

  results in a syntax error because it uses the less-than character (<), which is reserved for start tags and end tags such as <p> and </p>.
- HTML5 provides character entity references (in the form &code;) for representing

special characters (below figure). We could correct the previous line by writing

**<p>if x &lt10 then increment x by 1</p>**

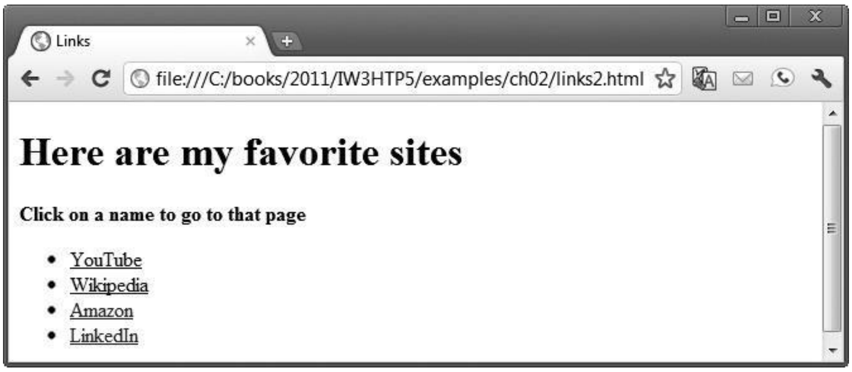
which uses the character entity reference &lt; for the less-than symbol (<). [Note: Before HTML5, the character entity reference &amp; was required to display an & in a web page. This is no longer the case.]



| Symbol | Description | Character entity reference |
|---|---|---|
| *HTML5 character entities* | | |
| & | ampersand | &amp; |
| ' | apostrophe | &apos; |
| > | greater-than | &gt; |
| < | less-than | &lt; |
| " | quote | &quot; |
| *Other common character entities* | | |
| non-breaking space | | &nbsp; |
| © | copyright | &copy; |
| — | em dash | &mdash; |
| – | en dash | &ndash; |
| ¼ | fraction 1/4 | &frac14; |
| ½ | fraction 1/2 | &frac12; |
| ¾ | fraction 3/4 | &frac34; |
| ... | horizontal ellipsis | &hellip; |
| ® | registered trademark | &reg; |
| § | section | &sect; |
| ™ | trademark | &trade; |

Figure: Some common HTML character entity references

## LISTS



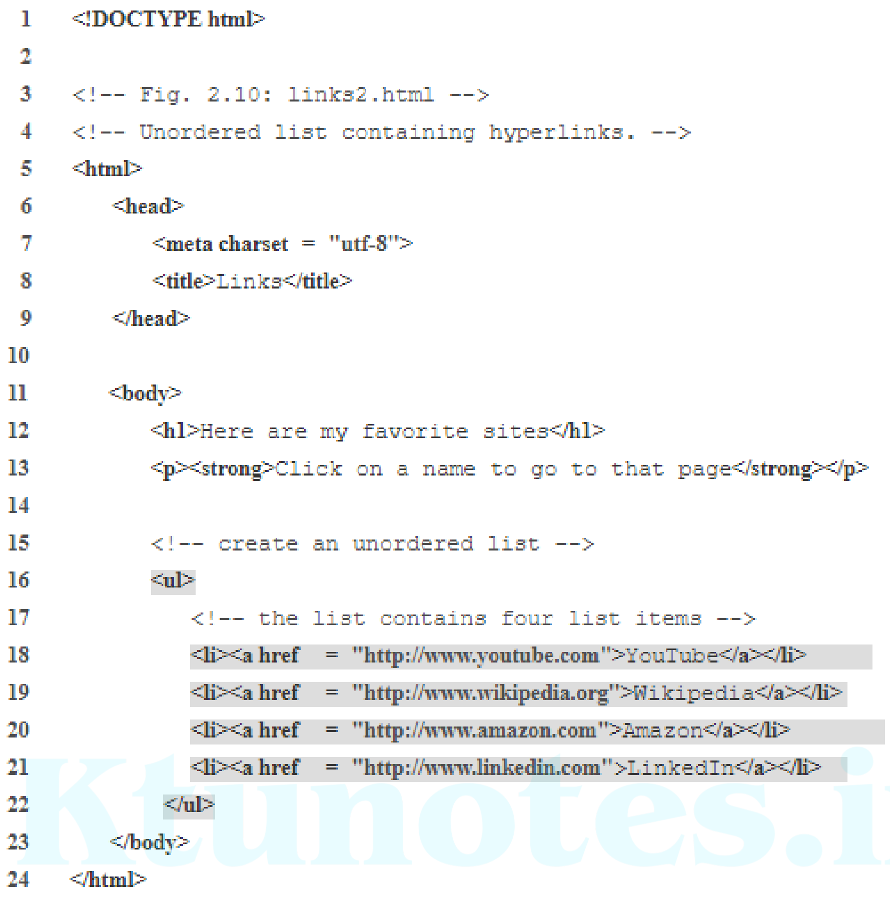
# Here are my favorite sites

**Click on a name to go to that page**

- YouTube
- Wikipedia
- Amazon
- LinkedIn

Figure.Unordered list containing hyperlinks.

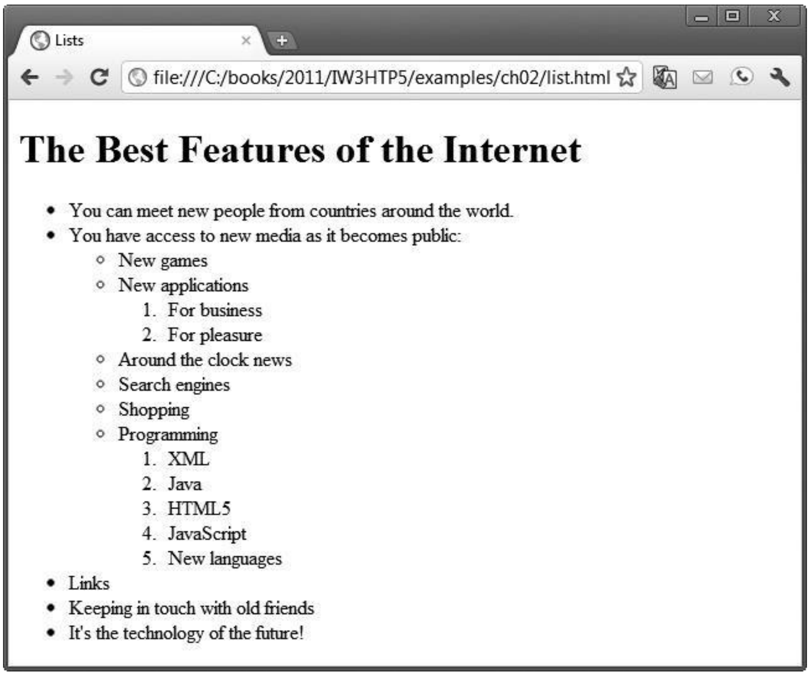
- Figure above displays text in an unordered list (i.e., a simple bullet-style list that does not order its items by letter or number). The unordered-list element ul (lines 16–22) creates a list in which each item begins with a bullet symbol (typically a disc). Each entry in an unordered list is an li (list item) element (lines 18–21). Most web browsers render each li element on a new line with a bullet symbol indented from the beginning of the line.



```
<!DOCTYPE html>

<!-- Fig. 2.10: links2.html -->
<!-- Unordered list containing hyperlinks. -->
<html>
    <head>
        <meta charset = "utf-8">
        <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 -->
        <ul>
            <!-- the list contains four list items -->
            <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>
```

Nested List

- Lists may be nested to represent hierarchical relationships, as in a multilevel outline. Figure below demonstrates nested lists and ordered lists. The ordered-list element ol creates a list in which each item begins with a number.
- In many browsers, the items in the outermost unordered list (lines 15–55) are preceded by discs. List items nested inside the unordered list of line 15 are preceded in many browsers by hollow circular bullets.
- A web browser indents each nested list to indicate a hierarchical relationship. The first ordered list (lines 29–33) includes two items. Items in an ordered list are enumerated 1., 2., 3. and so on.
- Nested ordered lists are enumerated in the same manner. Although not demonstrated in this example, subsequent nested unordered list items are often preceded by square bullets. The bullet styles used may vary by browser.



# The Best Features of the Internet

- You can meet new people from countries around the world.
- You have access to new media as it becomes public:
  - New games
  - New applications
    1. For business
    2. For pleasure
  - Around the clock news
  - Search engines
  - Shopping
  - Programming
    1. XML
    2. Java
    3. HTML5
    4. JavaScript
    5. New languages
- Links
- Keeping in touch with old friends
- It's the technology of the future!

Figure: Nested List and Ordered List

```
<!DOCTYPE html>

<!-- Fig. 2.11: list.html -->
<!-- Nested lists and ordered lists. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Lists</title>
   </head>

   <body>
      <h1>The Best Features of the Internet</h1>

      <!-- create an unordered list -->
      <ul>
         <li>You can meet new people from countries around
            the world.</li>
         <li>
            You have access to new media as it becomes public:

            <!-- this starts a nested unordered list, which uses a -->
            <!-- different bullet. The list ends when you -->
            <!-- close the <ul> tag. -->
            <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>
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>
```

## TABLES

Tables are frequently used to organize data into rows and columns. Example (Fig. below) creates a table with six rows and two columns to display price information for various fruits.

Figure: **Creating a basic table.**

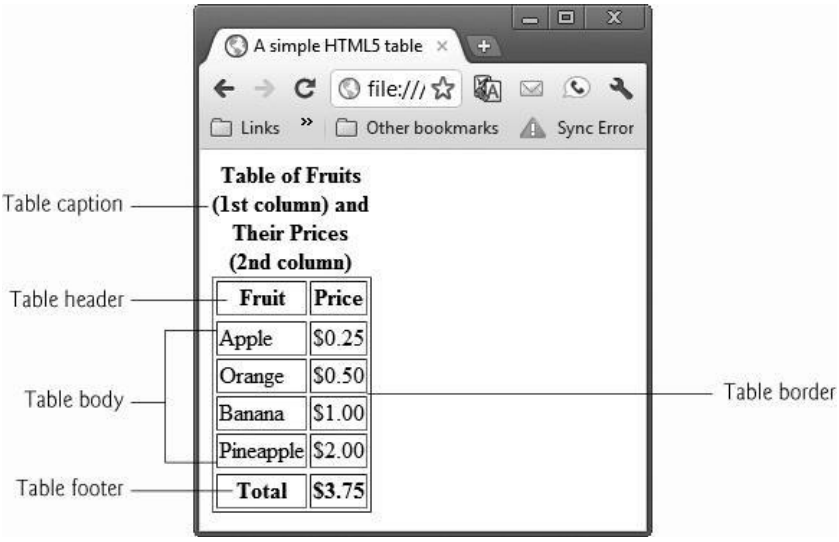
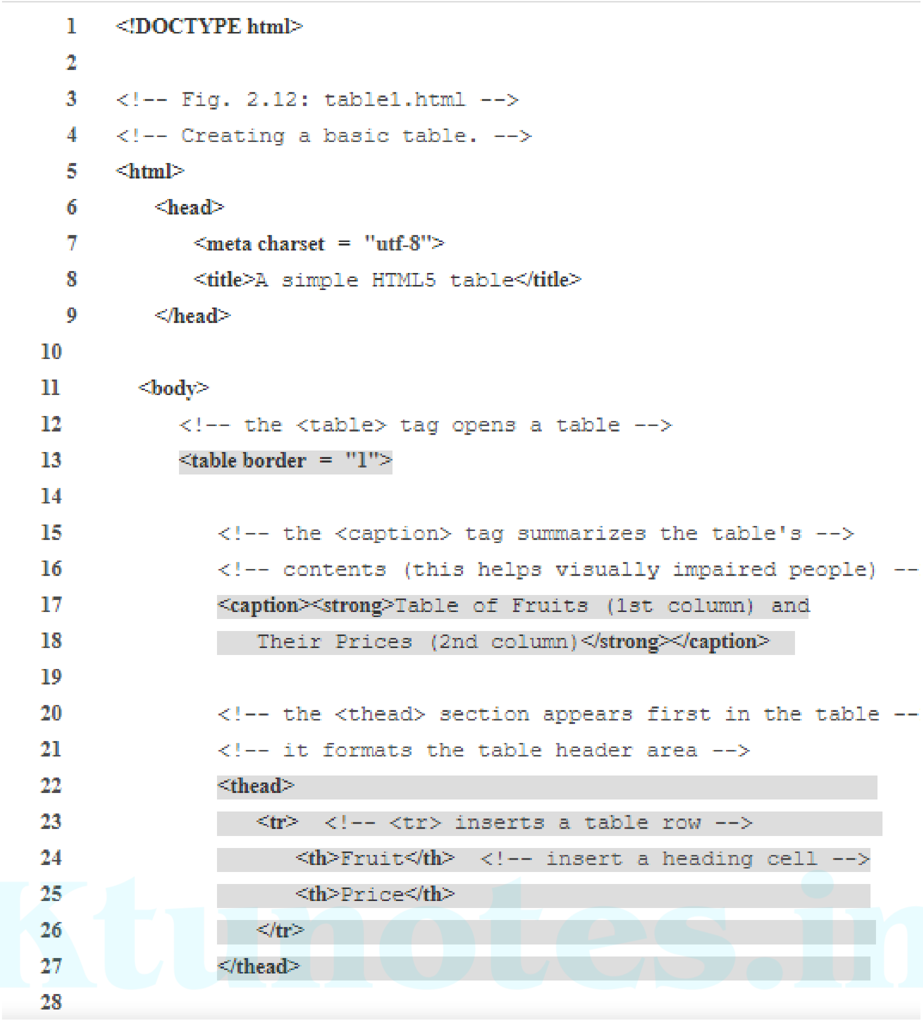
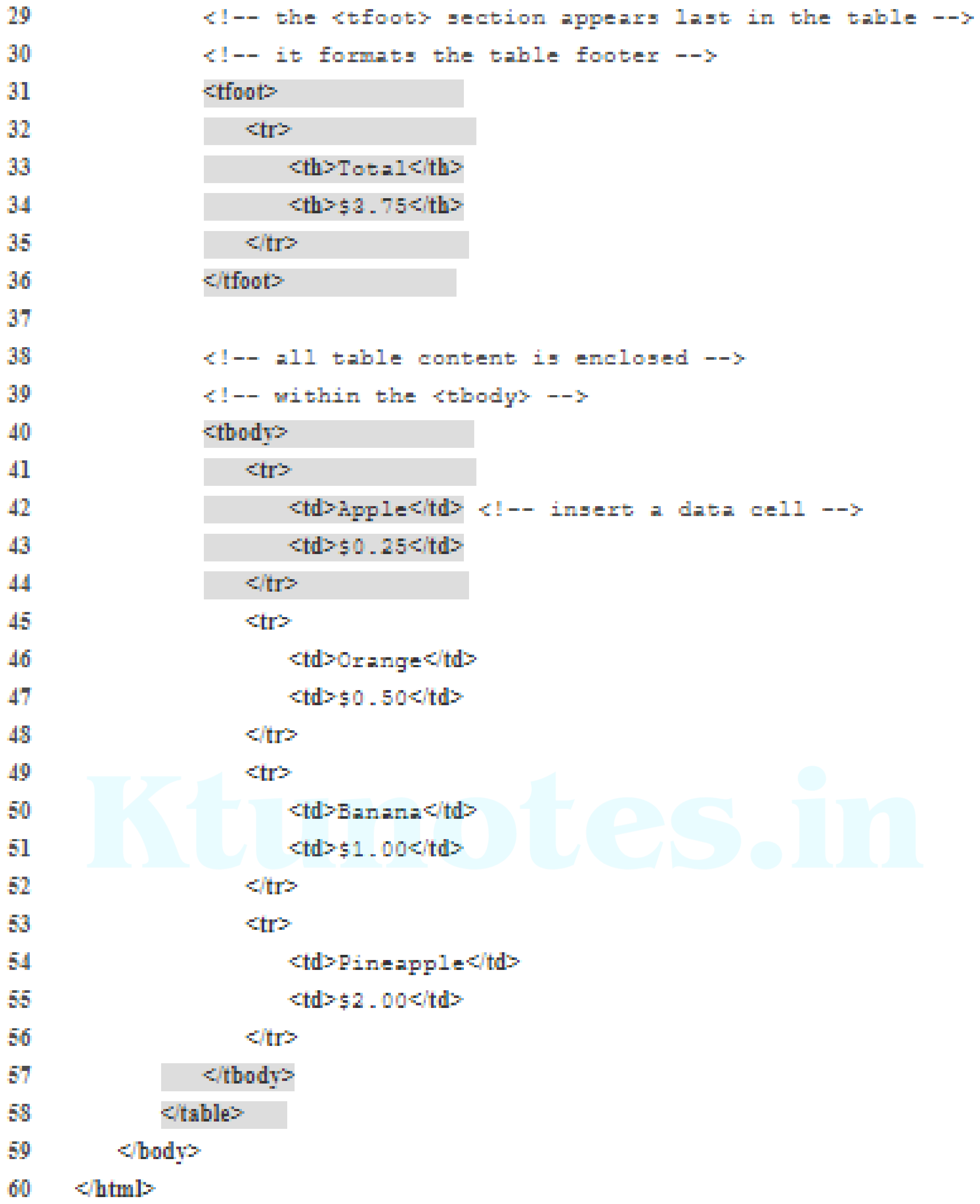


**Table of Fruits (1st column) and Their Prices (2nd column)**

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



```
<!DOCTYPE html>

<!-- Fig. 2.12: table1.html -->
<!-- Creating a basic table. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>A simple HTML5 table</title>
   </head>

  <body>
     <!-- the <table> tag opens a table -->
     <table border = "1">

        <!-- the <caption> tag summarizes the table's -->
        <!-- contents (this helps visually impaired people) --
        <caption><strong>Table of Fruits (1st column) and
           Their Prices (2nd column)</strong></caption>

        <!-- the <thead> section appears first in the table --
        <!-- it formats the table header area -->
        <thead>
           <tr>  <!-- <tr> inserts a table row -->
              <th>Fruit</th>  <!-- insert a heading cell -->
              <th>Price</th>
           </tr>
        </thead>
```

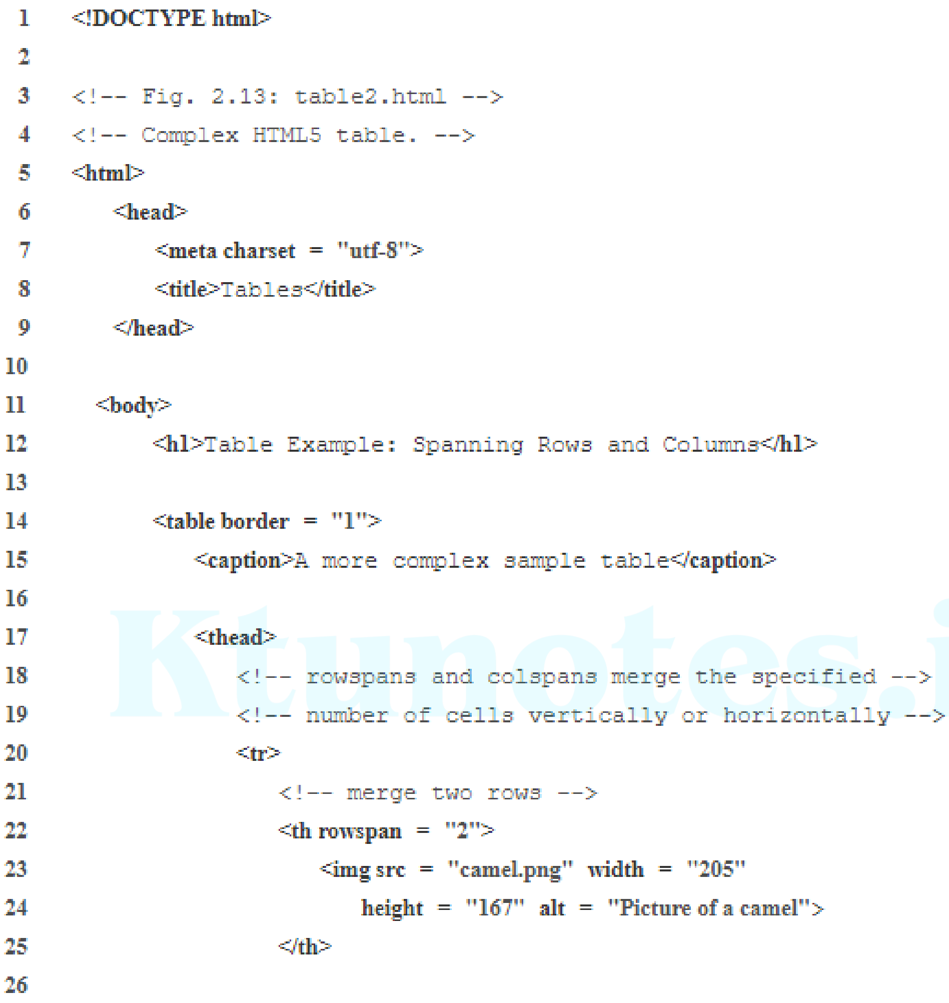


```
29            <!-- the <tfoot> section appears last in the table -->
30            <!-- it formats the table footer -->
31            <tfoot>
32              <tr>
33                <th>Total</th>
34                <th>$3.75</th>
35              </tr>
36            </tfoot>
37
38            <!-- all 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>
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>
```

Using rowspan and colspan with Tables

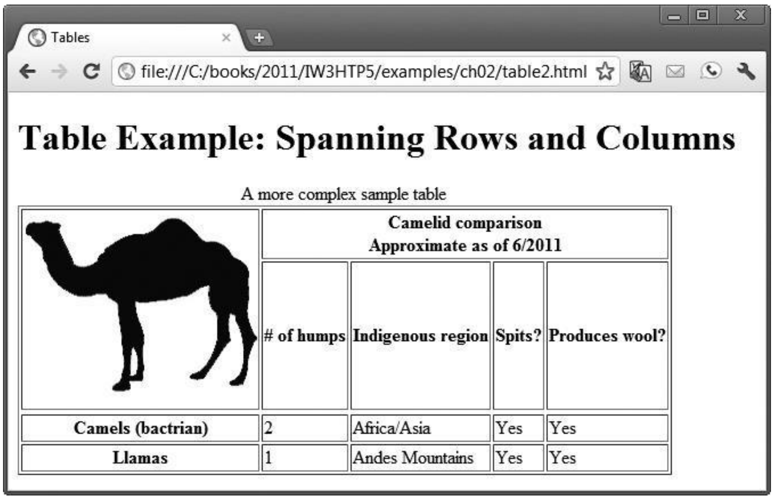
- The table begins in line 14. Table cells are sized to fit the data they contain, but you can control a table’s formatting using CSS3. We can create cells that apply to more than one row or column using the attributes rowspan and colspan. The values assigned to these attributes specify the number of rows or columns occupied by a cell. The th element at lines 22–25 uses the attribute rowspan = "2" to allow the cell containing the picture of the camel to use two vertically adjacent cells (thus the cell spans two rows). The th element in lines 28–31 uses the attribute colspan = "4" to widen the header cell (containing Camelid comparison and Approximate as of 10/2011) to span four cells.
- Line 29 introduces the br element, which most browsers render as a line break. Any

markup or text following a br element is rendered on the next line, which in this case appears within the same four-column span. 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.



```
<!DOCTYPE html>

<!-- Fig. 2.13: table2.html -->
<!-- Complex HTML5 table. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Tables</title>
   </head>

  <body>
      <h1>Table Example: Spanning Rows and Columns</h1>

      <table border = "1">
         <caption>A more complex sample table</caption>

         <thead>
            <!-- rowspans and colspans merge the specified -->
            <!-- number of cells vertically or horizontally -->
            <tr>
               <!-- merge two rows -->
               <th rowspan = "2">
                  <img src = "camel.png" width = "205"
                     height = "167" alt = "Picture of a camel">
               </th>

               <!-- merge four columns -->
               <th colspan = "4">
                  <strong>Camelid comparison</strong><br>
                  Approximate as of 10/2011
               </th>
            </tr>
            <tr>
               <th># of humps</th>
               <th>Indigenous region</th>
               <th>Spits?</th>
               <th>Produces wool?</th>
            </tr>
         </thead>
         <tbody>
            <tr>
               <th>Camels (bactrian)</th>
               <td>2</td>
               <td>Africa/Asia</td>
               <td>Yes</td>
               <td>Yes</td>
            </tr>
            <tr>
               <th>Llamas</th>
               <td>1</td>
               <td>Andes Mountains</td>
               <td>Yes</td>
               <td>Yes</td>
            </tr>
         </tbody>
      </table>
   </body>
</html>
```

Figure below represents another example and introduces new attributes that allow you to build more complex tables.



# Table Example: Spanning Rows and Columns

A more complex sample table

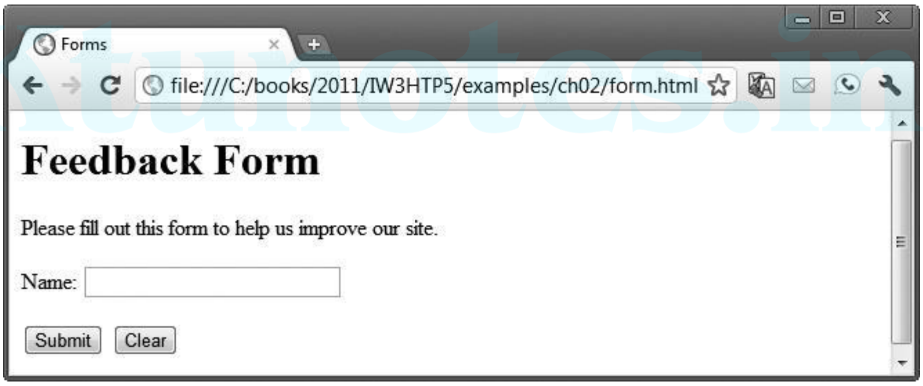
|  | Camelid comparison<br>Approximate as of 6/2011 |  |  |  |
|---|---|---|---|---|
|  | # of humps | Indigenous region | Spits? | Produces wool? |
| Camels (bactrian) | 2 | Africa/Asia | Yes | Yes |
| Llamas | 1 | Andes Mountains | Yes | Yes |

Figure.Complex HTML5 table.

## FORMS

- When browsing websites, users often need to provide information such as search queries, e-mail addresses and zip codes. HTML5 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, HTML5 documents, images, animations, videos). 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). HTML5 documents are requested and transferred via the Hypertext Transfer Protocol (HTTP).

Figure below is a simple form that sends data to the web server for processing.



Feedback Form

Please fill out this form to help us improve our site.

Name:

Figure:Form with text field and hidden fields

- The web server typically returns a web page back to the web browser—this page often indicates whether or not the form's data was processed correctly

method Attribute of the form Element

The form is defined in lines 20–43 by a form element. Attribute method (line 20) 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 of the form Element

The action attribute in the form element in line 20 specifies the URL of a script on the web server that will be invoked to process the form's data. Since we haven't used server-side programming here, we set this attribute to http://www.deitel.com.Lines 24–43 define input elements that specify data to provide to the script that processes the form (also called the form handler). There are several types of input elements. An input's type is determined by its type attribute. This form uses a text input, a submit input, a reset input and three hidden inputs.

```
l   <!DOCTYPE html>
2
3   <!--Fig.2.1 :=arm.html-->
4   <!--Form w:.th a text field and h:.dcten f:.elds.-->
s   <html>
6      <head>
7         <meta charset = "utf-8">
8         <tide>Forms<ftitle>
9      <!head>
10
11     <body>
12        <hl>Feedback  Form<Jhl>
13
14        <p>Please fill out this form to help
15           us improve our site.<lp>
16
17        <!--this tag starts the the form, gives the-->
18        <!--method of sending in=ormat:.on and the-->
19        <!--location of the form-process:.ng scr:.pt -->
20        <form method  = "post" action  = "http://www.deitel.c.om">
21           <!--  h:.dden inputs contain non-visual-->
22           <!--information that will a:so be submitted-->
23           <input t)-pe = "hidden" name = "recipient"
24              ,·alue  = "deitel@deitel.com">
25           <input -pe = "hidden" name = "subject"
26              ,·alue = "Feedback Form">
```

### Hidden Inputs

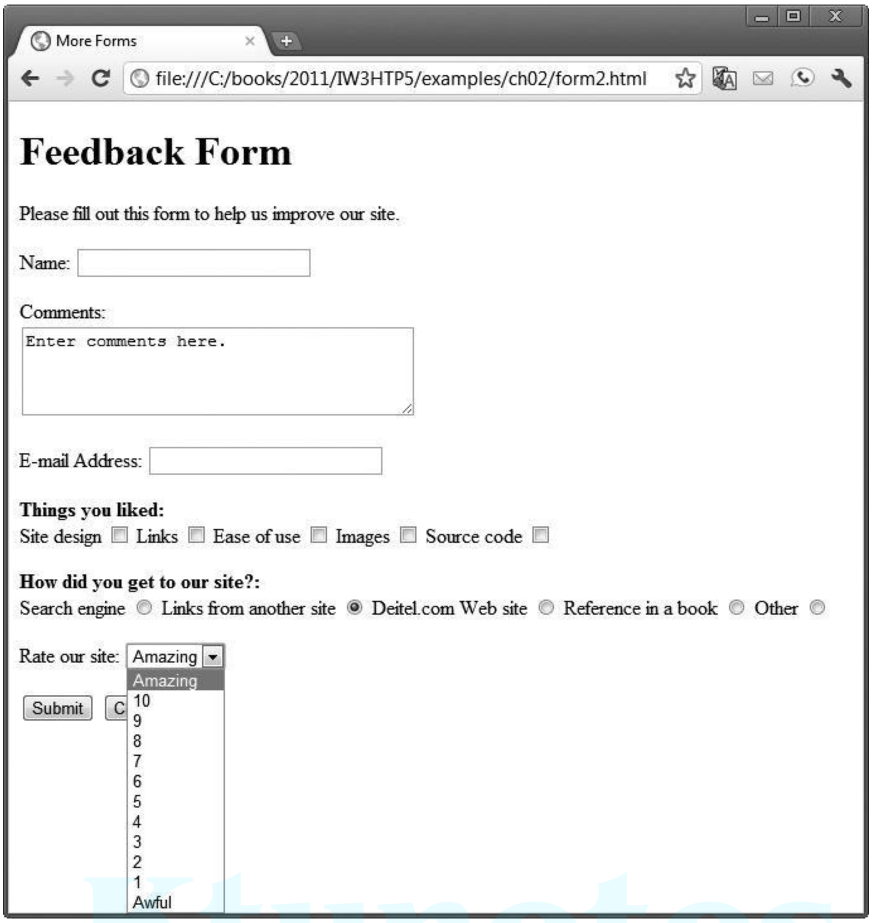
- Forms can contain visual and nonvisual components. Visual components include clickable buttons and other graphical user interface components with which users interact. Nonvisual components, called hidden inputs (lines 23–28), store any data that we specify, such as e-mail addresses and HTML5 document file names that act as links.
- The three hidden input elements in lines 23–28 have the type attribute hidden, which allows you to send form data that's not given by a user. The hidden inputs are an e-mail address to which the data will be sent, the e-mail's subject line and a URL for the browser to open after submission of the form. Two other input attributes are name, which identifies the input element, and value, which provides the value that will be sent (or posted) to the web server. The server uses the name attribute to get the corresponding value from the form.

### text input Element

The text input in lines 32–33 inserts a text field in the form. Users can type data in text fields. The label element (lines 31–34) 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 input Elements

Two input elements in lines 40–41 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).



## Feedback Form

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

### INTERNAL LINKING

- Figure below introduces internal linking—a mechanism that enables the user to jump between locations in the same document. Internal linking is useful for long documents that contain many sections. Clicking an internal link enables the user to find a section without scrolling through the entire document.

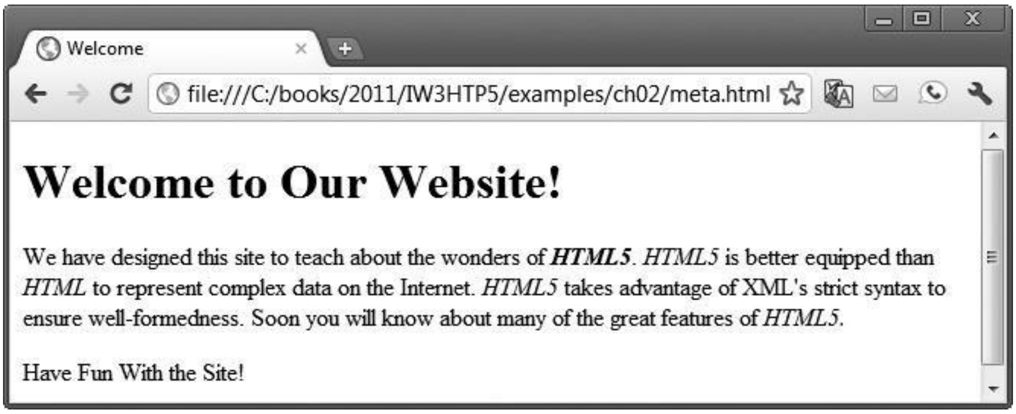
### META ELEMENTS

- Search engines catalog sites by following links from page to page (often known as spidering or crawling the site) and saving identification and classification information for each page.
- One way that search engines catalog pages is by reading the content in each page's meta elements, which specify information about a document. Using the meta element is one of many methods of search engine optimization (SEO)—the process of designing and tuning your website to maximize your findability and improve your rankings in organic (non-paid) search engine results.
- Two important attributes of the meta element are:

  ->name-which identifies the type of meta element

->content-which provides the information search engines use to catalog pages.

Figure below introduces the meta element.

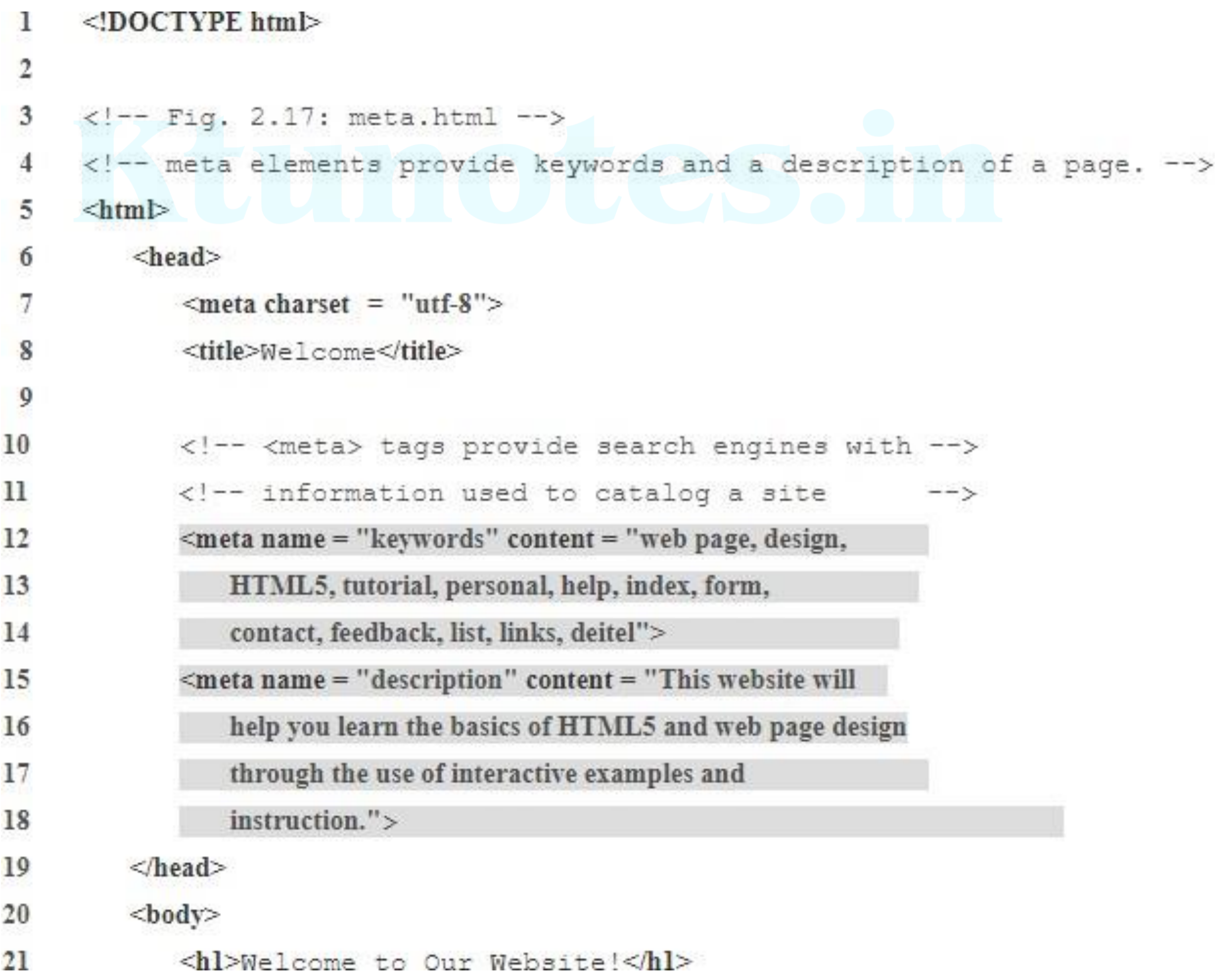


# Welcome to Our Website!

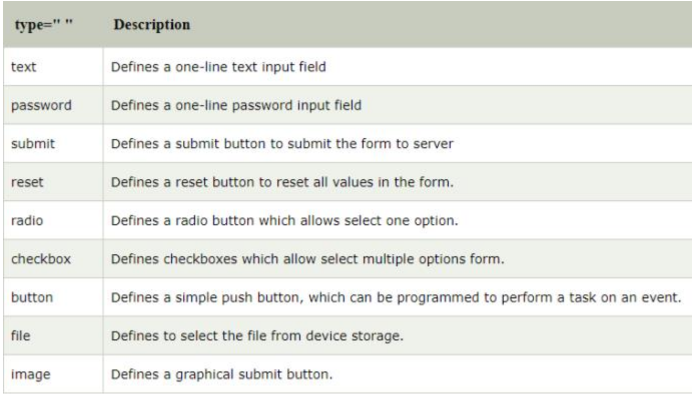
We have designed this site to teach about the wonders of ***HTML5***. *HTML5* is better equipped than *HTML* to represent complex data on the Internet. *HTML5* takes advantage of XML's strict syntax to ensure well-formedness. Soon you will know about many of the great features of *HTML5*.

Have Fun With the Site!

Figure: Meta elements provide keywords and a description of a page.



```
<!DOCTYPE html>

<!-- Fig. 2.17: meta.html -->
<!-- meta elements provide keywords and a description of a page. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Welcome</title>

      <!-- <meta> tags provide search engines with -->
      <!-- information used to catalog a site      -->
      <meta name = "keywords" content = "web page, design,
         HTML5, tutorial, personal, help, index, form,
         contact, feedback, list, links, deitel">
      <meta name = "description" content = "This website will
         help you learn the basics of HTML5 and web page design
         through the use of interactive examples and
         instruction.">
   </head>
   <body>
      <h1>Welcome to Our Website!</h1>
```



| type=" " | Description |
|---|---|
| text | Defines a one-line text input field |
| password | Defines a one-line password input field |
| submit | Defines a submit button to submit the form to server |
| reset | Defines a reset button to reset all values in the form. |
| radio | Defines a radio button which allows select one option. |
| checkbox | Defines checkboxes which allow select multiple options form. |
| button | Defines a simple push button, which can be programmed to perform a task on an event. |
| file | Defines to select the file from device storage. |
| image | Defines a graphical submit button. |

```
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>
```

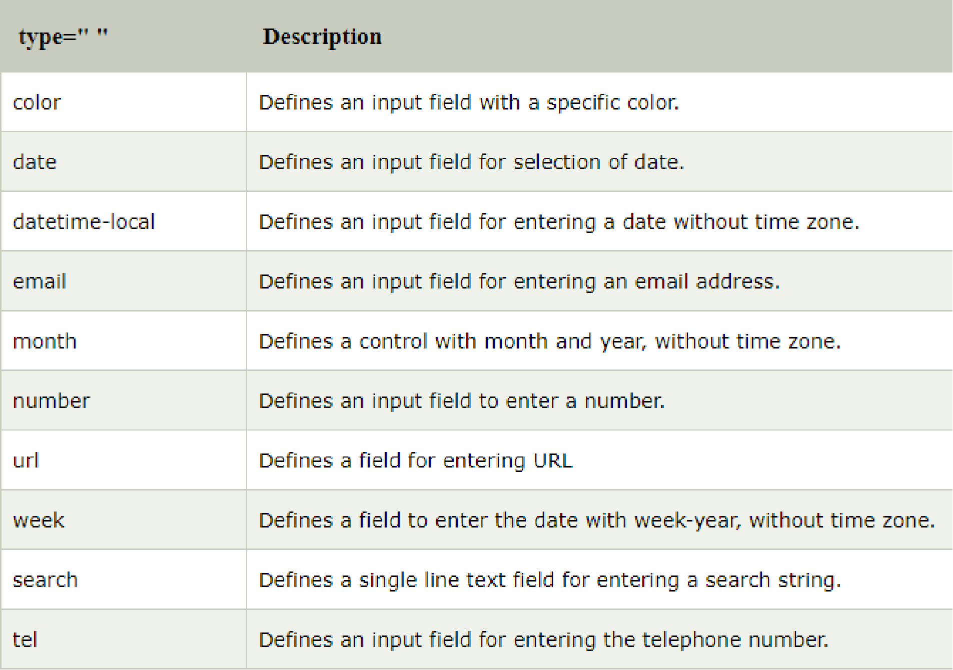
- Lines 12–14 demonstrate a "keywords" meta element. The content attribute of such a meta element provides search engines with a list of words that describe the page. These words are compared with words in search requests. Thus, including meta elements and their content information can draw more viewers to our site.
- Lines 15–18 demonstrate a "description" meta element. The content attribute of such a meta element provides a three- to four-line description of a site, written in sentence form. Search engines also use this description to catalog our site and sometimes display this information as part of the search results.

## NEW HTML5 FORM INPUT TYPES

- In HTML <input type=" "> is an important element of HTML form. The "type" attribute of input element can be various types, which defines information field. Such as <input type="text" name="name"> gives a text box.
- HTML5 introduces several new <input> types like email, date, time, color, range, and so on,to improve the user experience and to make the forms more interactive.
- However, if a browser failed to recognize these new input types, it will treat them like a normal text box.

**Following is a list of all types of <input> element of HTML**

**HTML5 added new types on <input> element. Following is the list of types of elements of HTML5:**



| type=" " | Description |
|---|---|
| color | Defines an input field with a specific color. |
| date | Defines an input field for selection of date. |
| datetime-local | Defines an input field for entering a date without time zone. |
| email | Defines an input field for entering an email address. |
| month | Defines a control with month and year, without time zone. |
| number | Defines an input field to enter a number. |
| url | Defines a field for entering URL |
| week | Defines a field to enter the date with week-year, without time zone. |
| search | Defines a single line text field for entering a search string. |
| tel | Defines an input field for entering the telephone number. |

**Following is the description about types of <input> element with examples:**

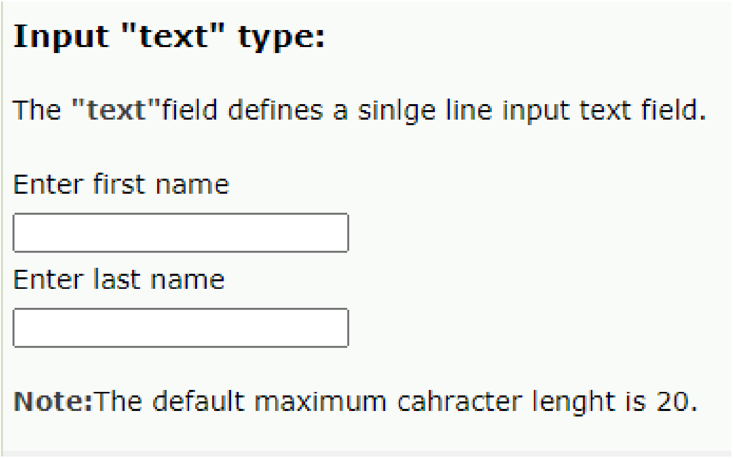
**1.** **<input type="text">:**

<input> element of type "text" are used to define a single-line input text field.

**E.g;**

```
<form>
    <label>Enter first name</label><br>
    <input type="text" name="firstname"><br>
    <label>Enter last name</label><br>
    <input type="text" name="lastname"><br>
    <p><strong>Note:</strong>The default maximum cahracter lenght
 is 20.</p>
</form>
```

**Output**



**Input "text" type:**

The "**text**"field defines a sinlge line input text field.

Enter first name

Enter last name

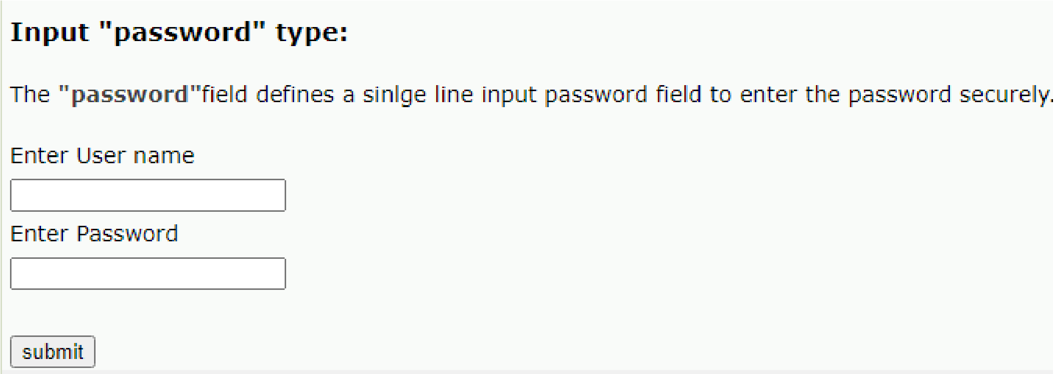
**Note:**The default maximum cahracter lenght is 20.

### 2. <input type="password">:

The <input> element of type "password" allow a user to enter the password securely in a webpage. The entered text in password filed converted into "*" or ".", so that it cannot be read by another user.E.g;

```
<form>
    <label>Enter User name</label><br>
    <input type="text" name="firstname"><br>
    <label>Enter Password</label><br>
    <input type="Password" name="password"><br>
    <br><input type="submit" value="submit">
</form>
```

**Output**



**Input "password" type:**

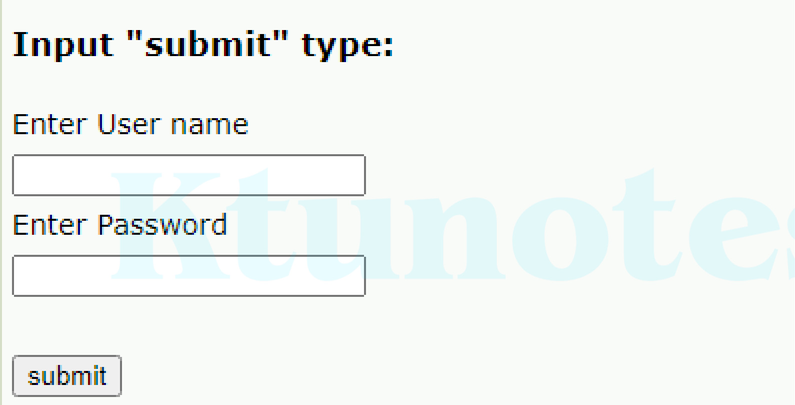
The **"password"**field defines a sinlge line input password field to enter the password securely.

### 3. <input type="submit">:

The <input> element of type "submit" defines a submit button to submit the form to the server when the "click" event occurs.E.g;

```
<form  action="https://www.javatpoint.com/html-tutorial">
    <label>Enter User name</label><br>
    <input type="text" name="firstname"><br>
    <label>Enter Password</label><br>
    <input type="Password" name="password"><br>
    <br><input type="submit" value="submit">
</form>
```

**Output**



Input "submit" type:

Enter User name

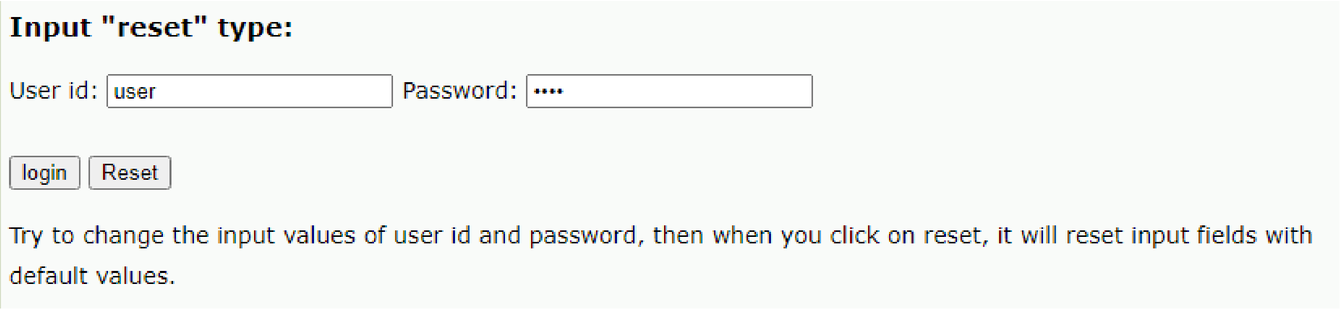
Enter Password

### 4. <input type="reset">:

The <input> type "reset" is also defined as a button but when the user performs a click event, it by default reset the all inputted values.E.g;

```
<form>
    <label>User id: </label>
     <input type="text" name="user-id" value="user">
              <label>Password: </label>
     <input type="password" name="pass" value="pass"><br><br>
     <input type="submit" value="login">
      <input type="reset" value="Reset">
</form>
```

**Output**



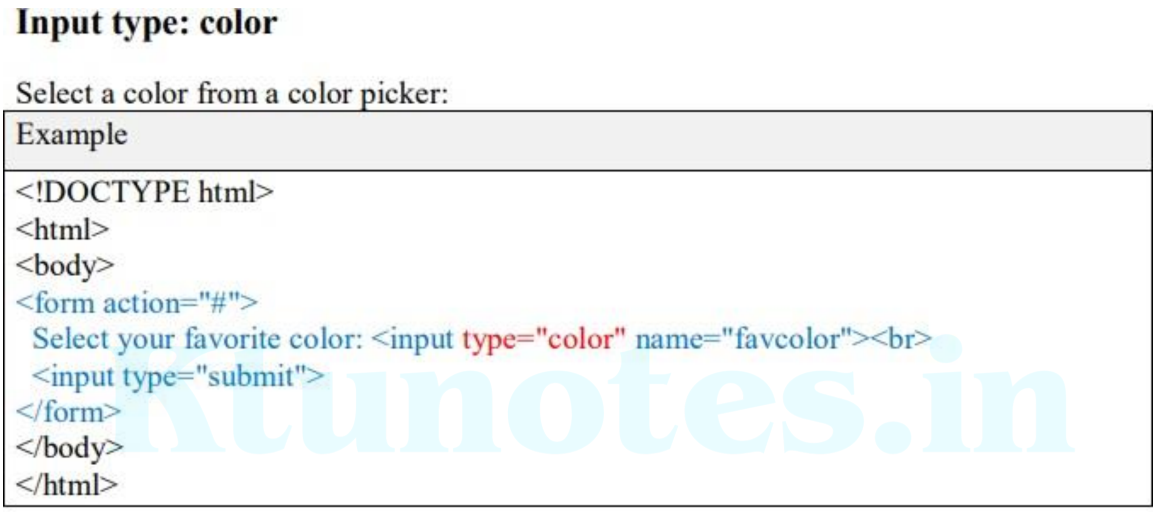
### Input "reset" type:

User id: user Password: ••••

Try to change the input values of user id and password, then when you click on reset, it will reset input fields with default values.

## HTML5 newly added <input> types element

https://www.javatpoint.com/html-form-input-types

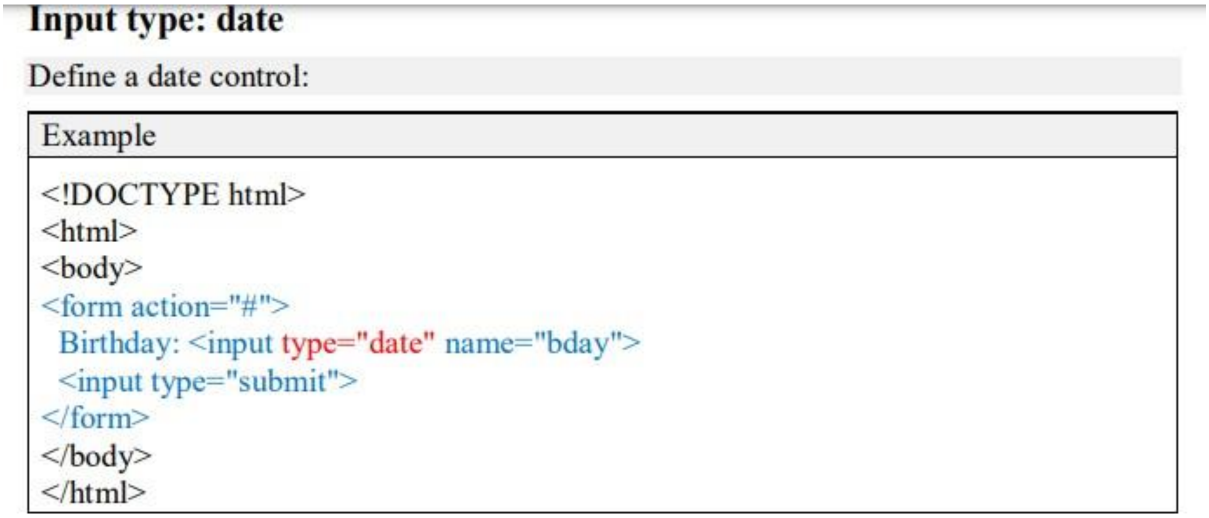


### Input type: color

Select a color from a color picker:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
  Select your favorite color: <input type="color" name="favcolor"><br>
  <input type="submit">
</form>
</body>
</html>
```



### Input type: date

Define a date control:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
  Birthday: <input type="date" name="bday">
  <input type="submit">
</form>
</body>
</html>
```

### Input type: datetime local

Define a date and time control (no time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
  Birthday (date and time): <input type="datetime-local" name="bdayt me">
  <input type="submit">
</form>
</body>
</html>
```

### Input type: email

Define a field for an e-mail addres (will be automatically alidated when submitted):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
  E-mail: <input type="email." name="usremail">
  <input type="submit">
</fonn>
</body>
</html>
```

### Input type: file

Define a file-select field and a 1'Browse...11 button (for file uploads):

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
  Select a file: <mnput type=' file" name="img">
  <input type="submit">
</fonn>
</body></html>
```

### Input type: hidden

Define a hidden field (not vi ible to a ser).
A hidden field o•ften stores a default a.lue or can ha e its val e changed by a

Example

```
<!DOCTYP   html>
<html>
<body>
<form achon="#">
  First name: <inpu   ype="te  t1, name="fname"><br>
  <input type="hidden"  name="country" value='    orway">
  <input type="submit"   alue="  ubmit1">
</form>
</badly>
</html>
```

fa a cript:

### Input type: image

Define an image as a submit button:

Example

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
  First name: <input type=" ext" name="fname"><br>
  Last name: <input type="te  t" name="lname"><br>
  <input type=1"image1, src=1"img_submit.gif' alt="  ubmit" width="48" height="48">
</form>
</body>
</html>
```

### Input type: month

Define a month aod year control (no time zone):

Example

```
<!DOCTYPE html>
<html>
<body>
<fo,m action="#">
  Birthday (month and year}: <input type="month11 name="bdaymonth">
  <input type="submit">
</fonn>
</body></html>
```

**Input type: number**

Define a field for entering a number (You can also set restrictions on what m:unbers are accepted}:

Example

```
<!DOCTYP, html>
<html>
<body>
<form action="#">
  Quantify (between 1 and 5): <inp t type=' number" name="quantity" min="l"
1na ="5">
  <input type="submitl'>
</fonn>
</body>
</html>
```

Use the following attributes to specify restrictions:

- max - specifies the maximum alue allo edl
- min - specifies the minimum value allowed
- step - spedfie the legal number interval
- value - Specifies the default value

**Input type: range**

Define a control for entering a number whose exact value is not important (like a slider control). You can al o set restrictions on what numbers are accepted:

Example

```
<!DOCTYPE html>
<html>
<body>
<form ac ion="#">
Points: <input type="range" name="points" min="O" max="10">
<input type="submit''>
</form>
</body>
</html>
```

Use ihe follo ing attribute to specify restrictions:

- max - specifies the maximum value allo e,dl
- min - specifies the mh1imum · alue a11o ed
- step - specifies the legal number inter als
- _alue - Specifi,e the default · alue

**Input type: reset**

Define a reset button re ets all form values to default values :

Ex.ample

```
<!DOCTYP    html>
<html>
<1body>
<fonn action-"#">
  Emai]: <input type"""email" name"""email"><br>
  Pin: <input typ    "t xt'1 name"""pin" maxlcngth--"4"><br>
  <input type="reset"    alu    "Re ct">
  <input type"""submit" valu    "Submit">
</form>
<,p>Click on the n;?:et button ro 11eset the form.</p>
<Jbody>
</html>
```

**Tip:** U ,e the reset butron ,carefuHyTIt ,can b annoying for u ers who :aeciden1tally act vate the reset butron.

**Input type; search**

Define a search field like a site se.arch, or Goo -le search :

Example

```
<!DOCTYPE html>
<html>
<1body>
<fonn action-"#">
  Search Goo le: <input type"""search" nam    "googl  carch"><br>
  <input type"""submit">
</form>
<Jbody>
</html>
```

**I■put type: tel**

Define a field for entering a telephone number:

Example

```
<!DOCTYPE html>
<html>
<1body>
<fonn actioo="#">
  Td ·phunc: <inp111 lypL="lel" narnc="11nld"><br_..
  <input type"""submit">
</form>
</body>
<fhitml>
```

**Input type: time**

Define a contml for entertn. a ttme , no time zone :

Example

```
<!DOCTYPE html>
<htnil>
<ibody>
<form action="#'1>
 Select a time: <input typ . "time" nam ,1u r time">
 <input typc="submit">
</form>
<  ody>
</html>
```

**Input type: url**

Define a field for entertng a l RL:

Ex.urnple

```
<!DOCTYPE html>
<htnil>
<1body>
<"form action="#'1.
 Add your homepage: <input typ ="url" namc="homepagc'1><br>
 <input typc="submit">
</form>
</body>
</html>
```

**Inp,nt type; week**

Define a week and year control (no ttme zone):

Example

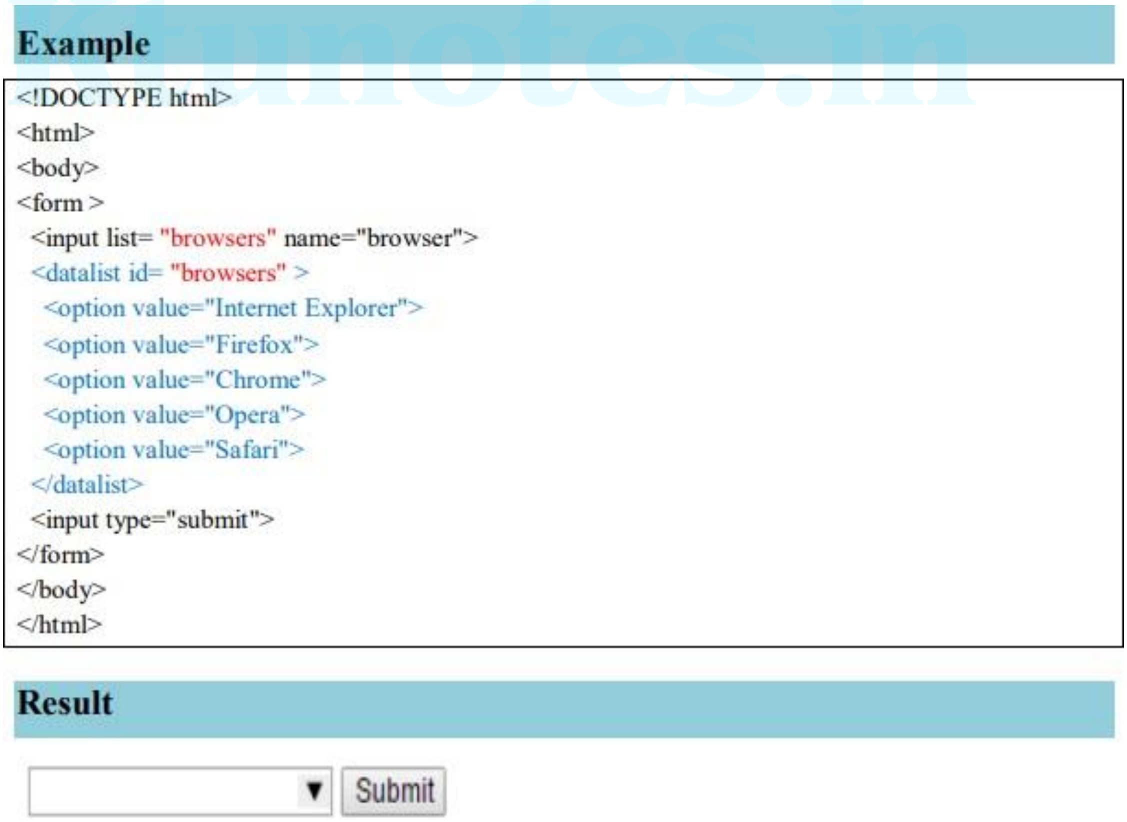
```
<!DOCTYPE html>
<html>
<1body>
<form action"""d mo_form.asp'1>
 Select a week: <input typ ="w ck" name="ycar_w ck">
 .(input type "submitr'>
</form>
</body>
</html>
```

## Input and Datalist Elements and Autocomplete Atttribute

- The autocomplete attribute(line 18) can be used on input types to automatically fill in the user’s information based on previous input. Such as name, address ,or email.
- We can enable autocomplete for an entire form or just for specific elements.
- E.g;An online order form might set autocomplete=”on” for the name and address inputs and set autocomplete=”off” for the credit card and password inputs for security purposes.

## Datalist Element

- The datalist element (lines 32-47) provides input options for a text input element. At the time of this writing,datalist support varies by browser.
- In the below example we use a datalist element to obtain the user’s birth month.
- The <datalist> tag is used to provide an "autocomplete" feature on <input> elements.
- Users will see a drop-down list of pre-defined options as they input data.
- Use the <input> element's list attribute to bind it together with a <datalist> element



Example

```
<!DOCTYPE html>
<html>
<body>
<form >
  <input list= "browsers" name="browser">
  <datalist id= "browsers" >
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit">
</form>
</body>
</html>
```

Result

## Page-Structure Elements

- HTML5 introduces several page structure elements that meaningfully identify areas of the page as headers,footers,navigation areas,asides,figures and more.

1. **Header Element**

   Creates a header for this page that contains both text and graphics.The header element can be used multiple times on a page and can include HTML headings(<h1 through h6>),navigation,images and logos and more.
   E.g;Top of front side of newspaper.

   **Title element**

   It does not need to be enclosed in a header,enables us to identify a date ,a time or both.

2. **Nav Element**

   This element groups navigation links.

   Given example used the heading "Recent Publications" and created a 'ul' element with seven 'li' elements that link to the corresponding web pages for each book.

3. **Figure element and Figcaption element**

   **Figure element**:Describes a figure(image,chart or table) in the document so that it could be moved to the side of the page or to another page.It doesnot include any styling but we can style the element using CSS.
   **Figcaption element**: Provides caption for the image in the figure element.

4. **Article element**

   Describes standalone content that could be potentially be used or distributed.E.g;news article,blog entry.

5. **Summary element and Details element**

   **Summary element-**displays a right pointing arrow next to a summary or caption when the document is rendered in browser.

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

6. **Section element**

   Describes a section of the document,usually with a heading for each section-these elements can be nested.

   E.g;We could have a section element for a book,then nested sections for each chapter name in the book.

   Given example "Recent Publications"-broke the document into 3 sections.

7. **Aside element**

   Describes the content that is related to the surrounding content.(an article),but is somewhat separate from the flow of text.

   E.g;An aside in a news story might include some background history.

8. **Meter element**

   Renders a visual representation of a measure within a range.

   E.g the total responses in the survey.min attribute is "0" and max attribute is "54". The value attribute is "14" representing the total number of people who responded "yes" to the survey question.

9. **Footer element**

   Describes a footer content that usually appears at the bottom of the content or section document.

10. **Text-Level Semantics:mark Element and wbr element** **Mark** :element highlights the text that is enclosed in the element.

    **Wbr**:element indicates the appropriate place to break a word when the text wraps to multiple lines.We use wbr to prevent a word from breaking in an awkward place.