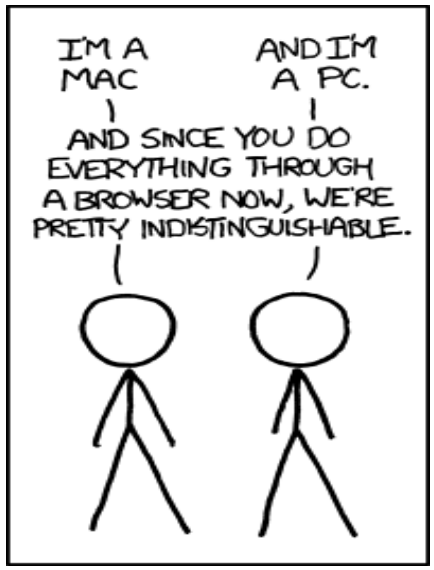
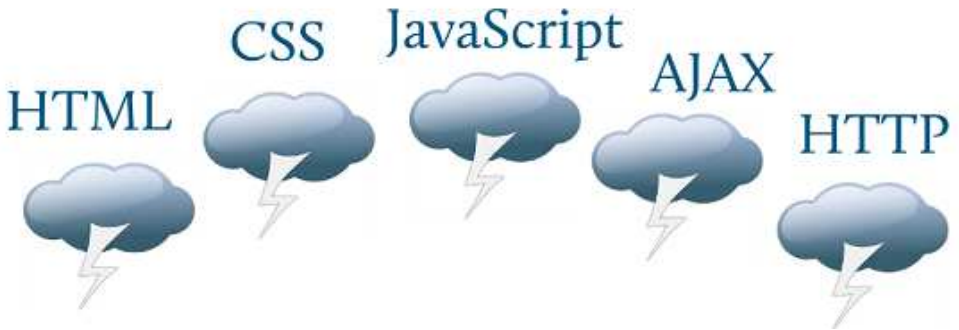
MIS6070

WEB BASED INFORMATION SYSTEMS





Lesson 3

HTML5 ≈ HTML 5 + CSS 3 + JavaScript

- HTML5 is a suite of tools for:
  - Markup (HTML 5)
  - Presentation (CSS 3)
  - Interaction (DOM, Ajax, APIs)
- Brought on by the evolving use of the web

How did HTML5 get Started?

- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0.
  - In 2006, they decided to cooperate and create a new version of HTML.
- Some rules for HTML5 were established:
  - New features should be based on HTML, CSS, DOM, and JavaScript
  - Reduce the need for external plugins (like Flash)
  - Better error handling
  - More markup to replace scripting
  - HTML5 should be device independent
  - The development process should be visible to the public

New Features

- Some of the most interesting new features in HTML5 :
  - The canvas element for drawing
  - The video and audio elements for media playback
  - Better support for local offline storage
  - New content specific elements, like article, footer, header, nav, section
  - New form controls, like calendar, date, time, email, url, search

Tag Pairs

- Keywords that help to give an HTML page structure
- Keyword is surrounded by angled brackets
- Most tags come in pairs
  - Opening or start tag
  - Closing or end tag

```
<h1>Pet Care 101</h1>
```

- Closing tag must have same case as opening tag

Empty Tags

- Empty tags don't require an end tag
- Examples:
  -
 for a line break
  - <hr /> for a horizontal line

Common HTML Tags

- <html>: Identifies the page as an HTML document
- <head>: Contains markup and code used by the browser, such as scripts that add interactivity, and keywords to help search engines find the page
- <title>: Displays the title of the Web page, which appears at the top of the Web browser, usually on the page's tab in a tabbed browser
- <body>: Surrounds content that's visible on the Web page when viewed in a Web browser

Common HTML Tags (Continued)

- <a href=URL>: Generally used to anchor a URL to text or an image; can create a named anchor within a document to allow for linking to sections of the document
- <b>: Applies boldface to text
- <hx>: Creates a heading, which can be first level (h1) through sixth level (h6)
- <img>: Inserts an image from a file or another Web site
- <p>: Defines text as a paragraph

Required HTML Tags

- Tags required on every Web page:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>    </title>
        <meta charset="UTF-8">
      <style>   </style>     // for header and article and img
      <script>   </script>
    </head>
    <body>
      content: header, article for each site
    </body>
    </html>
```

Elements

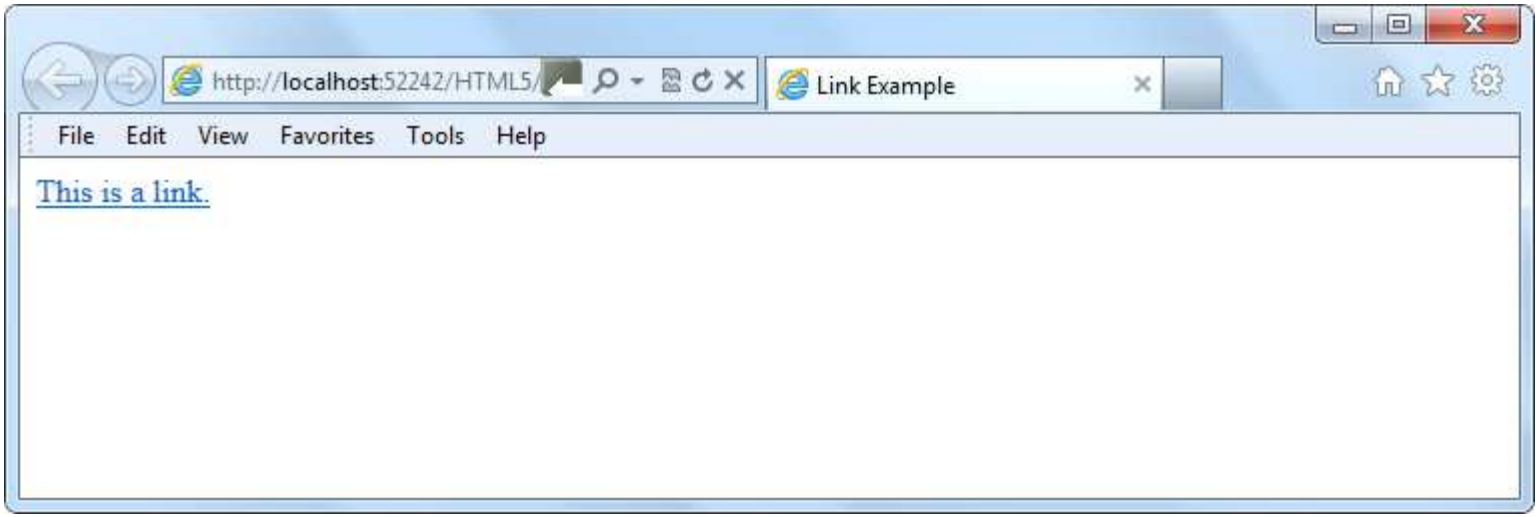
- A tag pair or an empty tag is also called an **element.**
- An element can describe content, insert graphics, and create hyperlinks.

Attributes

- Modifiers of HTML elements that provide additional information
- Are extensions of elements
- Syntax: `<tag attribute="value">`

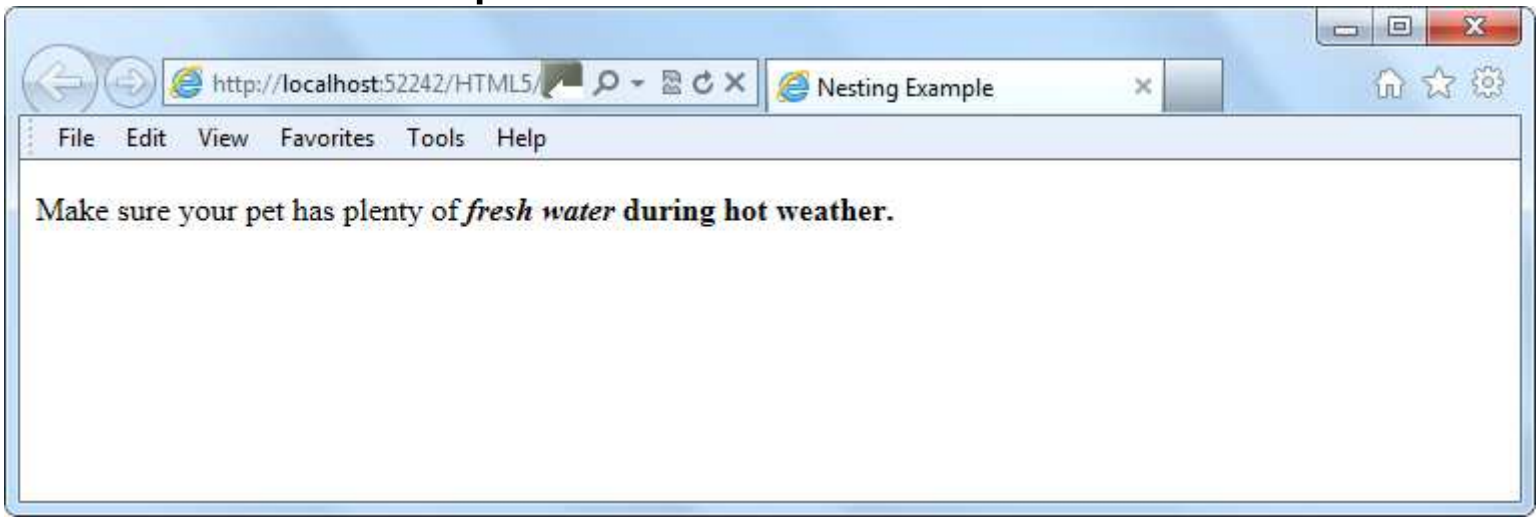
Creating a Link

- Example:
  - `<a href="http://www.example.com">This is a link.</a>Global attribute`



Nesting

- To place one element inside another
- Example:
  - `<p>Make sure your pet has plenty of <i><b>fresh water</i> during hot weather.</p></b>`



Entities

- A special character, such as the dollar symbol, the registered trademark (a capital R within a circle), and accented letters
- Begins with an ampersand (&) and ends with a semicolon (;)
- Examples:
  - entity ® represents the registered trademark symbol
  - Its numerical code is ®

Entities (Continued)

- Called character encoding
- Use UTF-8 encoding whenever possible
- Add the following declaration to the head element:

```
<meta charset="UTF-8">
```

Doctype

- A declaration found at the very top of almost every HTML document
- Specifies the language or rules the page uses

Doctype

- HTML 4.01 doctype example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.1//EN"
"http://www.example.com/TR/xhtml11/
DTD/xhtml11.dtd">
```

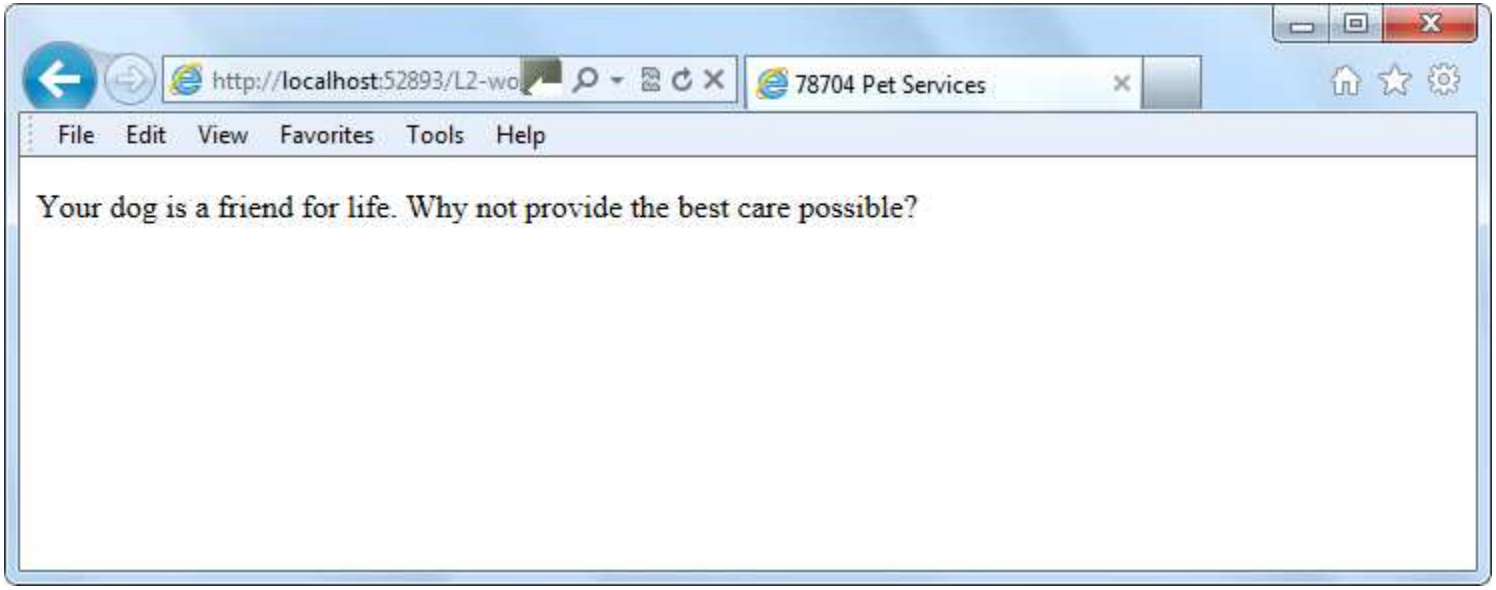
- HTML5 doctype:

```
<!doctype html>
```

Simple Web Page Markup Example

```
<!doctype html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>78704 Pet Services</title>
  </head>

<body>
  <p>Your dog is a friend for life. Why not
  provide the best care possible?</p>
</body>

</html>
```

Simple Web Page Markup Example



HTML 4 Text-related Elements with New Meaning or Functionality in HTML5

- The <b> element should now be used to offset text without conveying importance, such as for keywords or product names.
- The <i> element now indicates content in an alternate voice or mood, like spoken text.
- The element indicates strong importance.
- The element indicates emphatic stress.
- The <small> element should be used for small print, like a copyright line.

New HTML5 Elements

- The <command> element creates a command button. When the user clicks a command button, a command executes.
- The <mark> element highlights text on a page, similar to the highlighting feature in Microsoft Word.
- The <time> element displays a machine-readable time and date, such as 10:10 A.M., CST, July 19, 2012, which is handy for blogs and calendars, and potentially helps search engines provide better results when time and date are part of the search criteria.

Deprecation

- While new elements become available, the W3C earmarks other elements for eventual removal because their functionality is no longer useful.
- Removing elements from the list of available HTML elements is referred to as **deprecation.**
- **Note:** The same thing applies to attributes and CSS properties.

Examples of Deprecated HTML Elements

- `<big>`: Makes text bigger relative to the current font size
- `<center>`: Center-aligns text and content

- The fix: Use CSS instead

Image Basics

- A **raster image** is made up of pixels.
  - Example: A photograph
  - Formats: JPG, PNG, GIF, BMP
- A **vector image** is made up of lines and curves based on mathematical expressions.
  - Example: Adobe Illustrator AI file
  - Formats: PNG or GIF for Web display

The `img` element

- Use `img` to add images to an HTML document
- Example: `<img src="images/redball.jpg" alt="Red ball graphic" />`
- The `src` attribute and the `alt` attribute are required to be fully valid.
- The W3C requires the alt attribute for accessibility by people with disabilities.

`figure` and `figcaption` Elements

- The `figure` element specifies the type of figure you're adding
- The `figcaption` element adds a caption to an image on a Web page
  - Can display the caption before or after the image

figure and figcaption Example

```
<figure>
  <img src="doghappy.jpg" alt="Happy dog"
  width="100" height="125" />
  <figcaption>Happy dogs are good dogs</figcaption>
</figure>
```

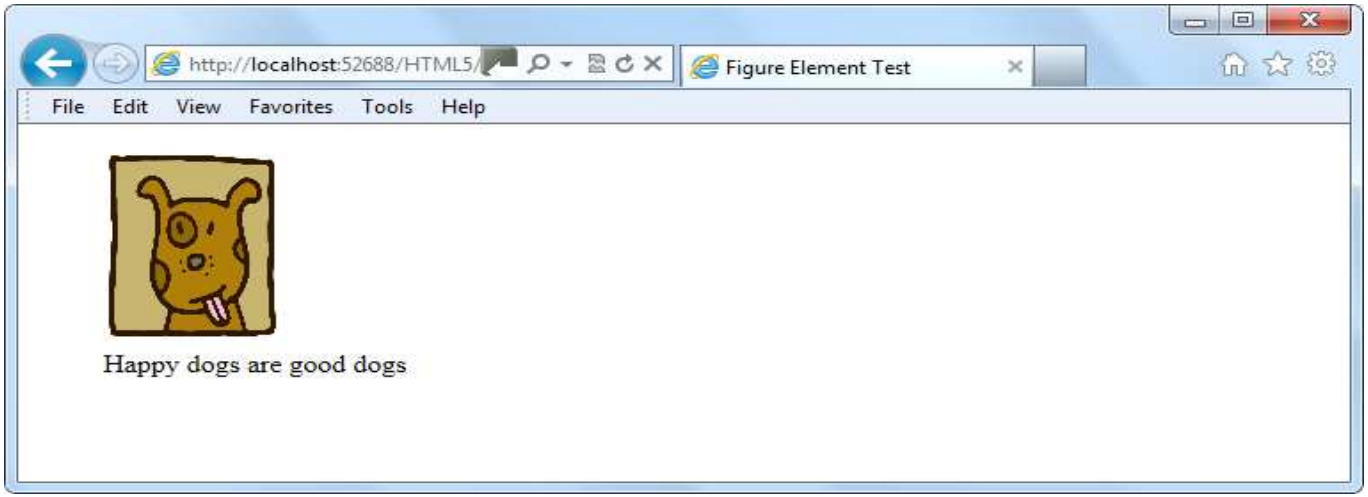
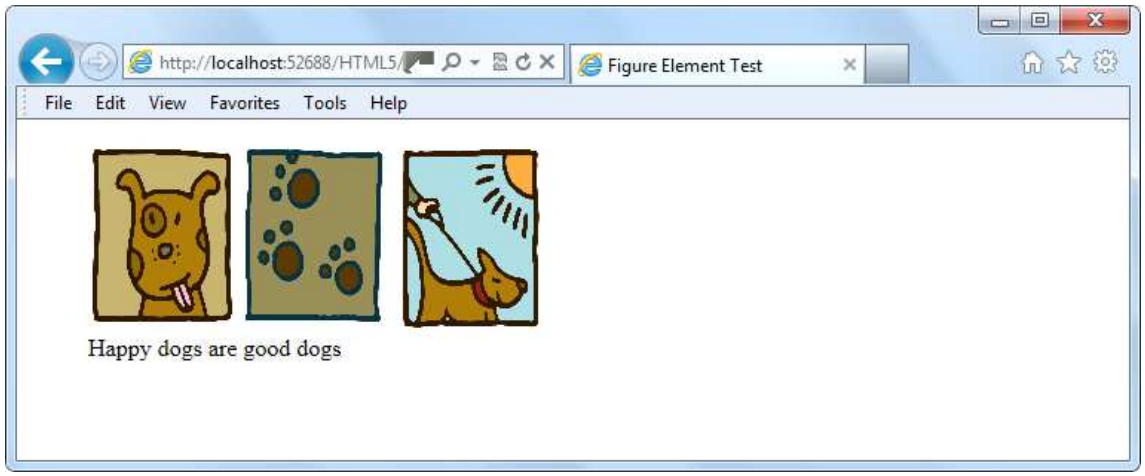


Illustration: © MightyIsland/iStockphoto

Side by Side Example

```
<figure>
  <img src="doghappy.jpg" alt="Happy dog"
  width="100" height="125" />
  <img src="dogpaws.jpg" alt="Dog paws"
  width="100" height="125" />
  <img src="dogwalk.jpg" alt="Walking a dog"
  width="100" height="125" />
  <figcaption>Happy dogs are good dogs</figcaption>
</figure>
```



Illustrations: © MightyIsland/iStockphoto

Canvas

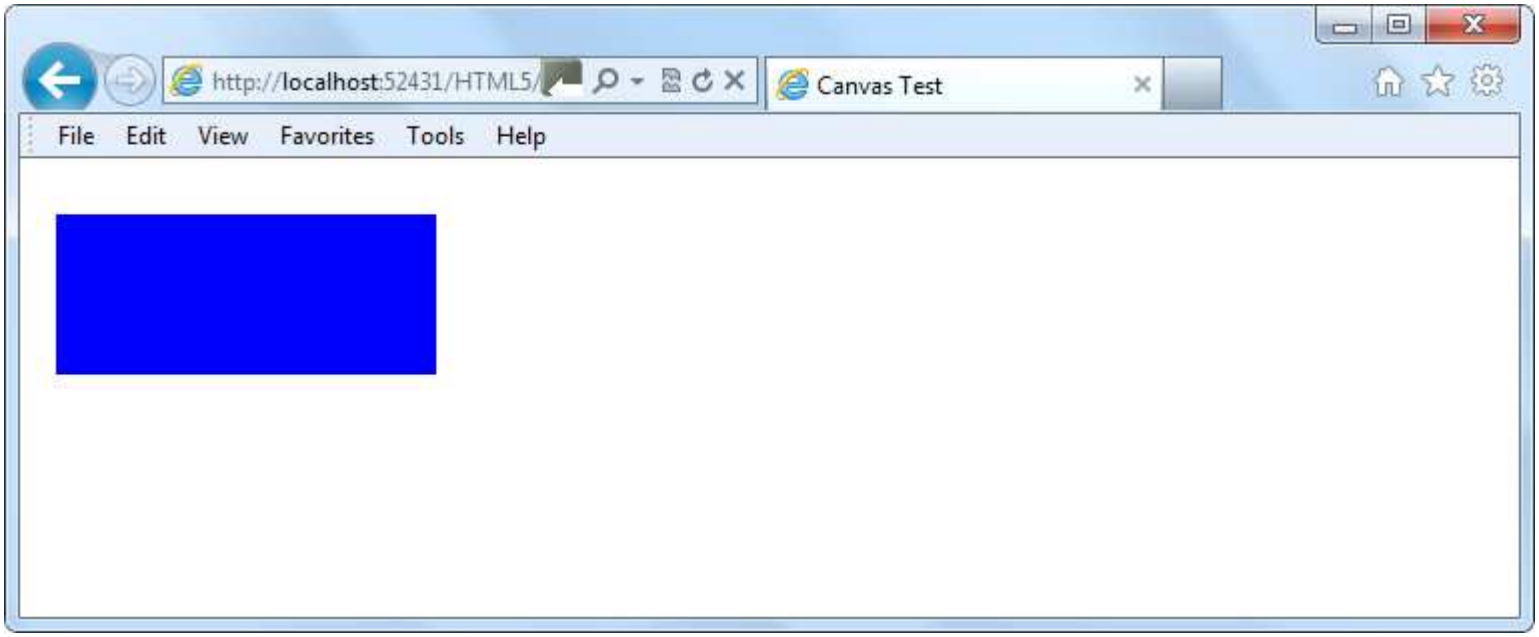
- Use JavaScript to draw pixel-based shapes on a canvas
- Include color, gradients, and pattern fills
- Render text with various embellishments
- Animate objects by making them move, change scale, and so on
- Basic syntax for the canvas element:

```
<canvas id="smlRectangle" height="100"
width="200"></canvas>
```

Canvas Example

```
<!doctype html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Canvas Test</title>

<script>
   function f1() {
       var canvas =
       document.getElementById("smlRectangle");
       context = canvas.getContext("2d");
       context.fillStyle = "rgb(0,0,255)";
       context.fillRect(10, 20, 200, 100);
   }
  </script>
  </head>

<body onload = "f1();">

<canvas id="smlRectangle" height="100" width="200 ">
</canvas>

</body>

</html>
```

Canvas Example



Fallback

- “Backup” content that displays if primary content cannot
- Can be a problem with some older browsers
  - Cannot render canvas drawings or animation, for example
- Fallback adds an image, text, or some other HTML content within the canvas element that displays if the drawing cannot

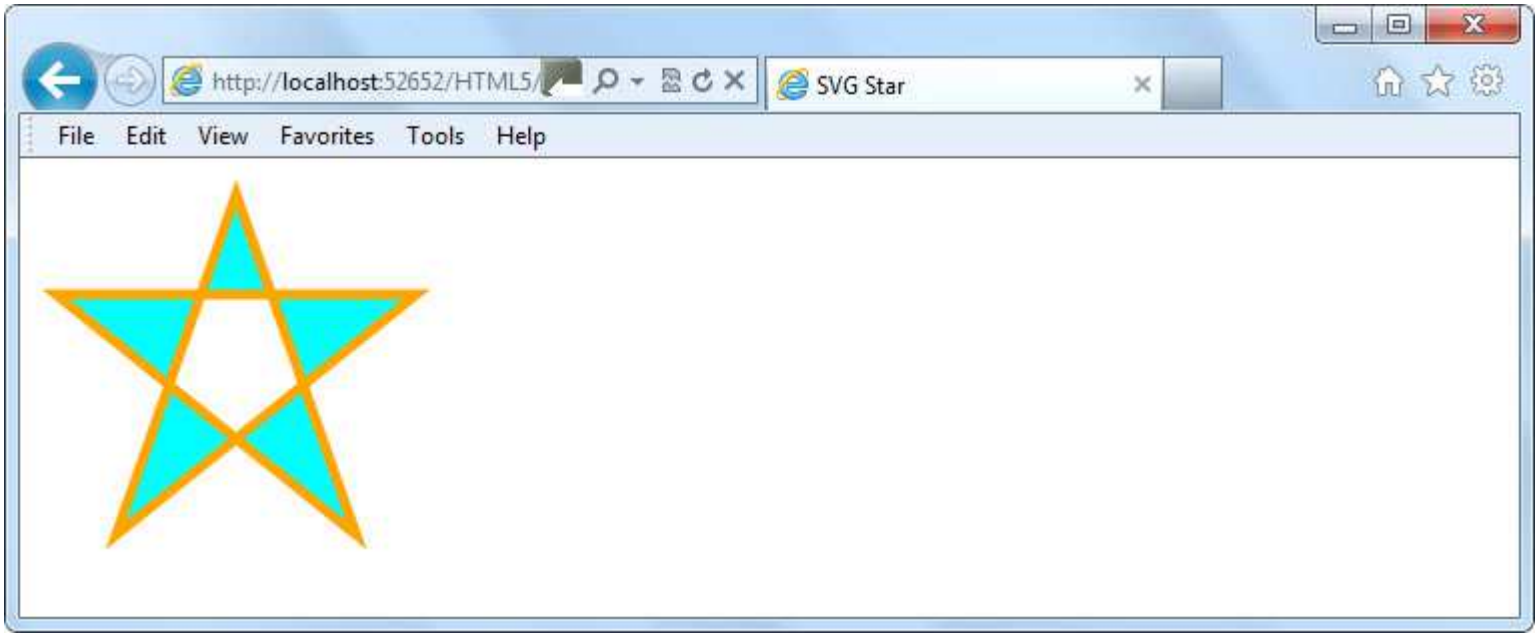
Scalable Vector Graphics (SVG)

- A language for describing 2D vector graphics in Extensible Markup Language (XML)
- SVG graphics referred to as objects
- SVG loads into the DOM
- Vector graphic changes size to fit screen, whether 32-inch PC monitor or smartphone
- SVG is not new, but HTML5 can embed SVG objects in Web pages without using `<object>` or `<embed>` tags

SVG Example

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>SVG Star</title>
  </head>
<body>
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <polygon points="100,10 40,180 190,60 10,60 160,180"
style="fill:aqua;stroke:orange;stroke-width:5;
fill-rule:evenodd;"/>
</svg>
</body>
</html>
```

SVG Example


http://localhost:52652/HTML5/
SVG Star
File Edit View Favorites Tools Help

When to Use Canvas Instead of SVG

- If the drawing is relatively small, use canvas.
- If the drawing requires a large number of objects, use canvas; SVG degrades as it continually adds objects to the DOM
- Generally, use canvas for small screens and SVG for larger screens.
- If you must create highly detailed vector documents that must scale well, go with SVG.
- If you are displaying real-time data output, such as maps, map overlays, weather data, and so on, use canvas.

`video` Element

- Enables you to incorporate videos in HTML documents using minimal code
- Markup example:

```
<video src="intro.mp4" width="400"
height="300">
</video>
```

`video` Attributes

- `poster`: Displays a static image file before the video loads
- `autoplay`: Start playing the video automatically upon page load
- `controls`: Displays a set of controls for playing, pausing, and stopping the video, and controlling the volume
- `loop`: Repeats the video

Video Markup Example

```
<video
  width="400" height="300"
  poster="78704-splash.jpg"
  autoplay="autoplay"
  controls="controls"
  loop="loop">

  <source src="intro.mp4" type="video/mp4" />

  </video>
```

`audio` Element

- Enables you to incorporate audio (music, other sounds) in HTML documents using minimal code
- Markup example:

```
<audio src="sample.mp3"
controls="controls">
</audio>
```

audio Example

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Audio Test</title>
  </head>
<body>
  <audio src="sample.mp3" controls="controls">
  </audio>
</body>
</html>
```

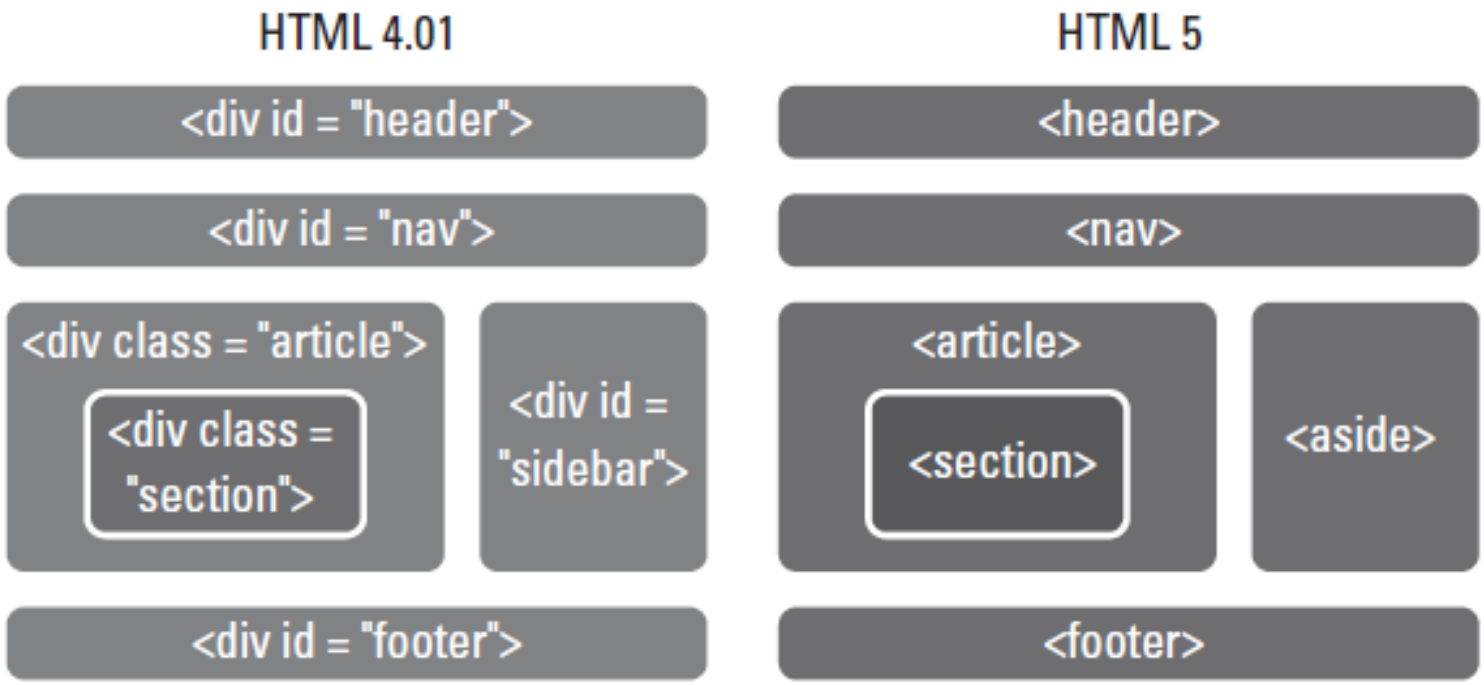
`audio` Example

`div` Element

- Used for years to create structure of an HTML document
- Often includes a **class** or **ID attribute**
- May include CSS styles such as `background-color`, `height`, and `width`
- Example:
  - `<div id="header" > This is a header </div>`

New HTML5 Elements for Structuring and Organizing Content

- `header`, `footer`, `section`, `nav`, `article`, and `aside`



Semantic Markup

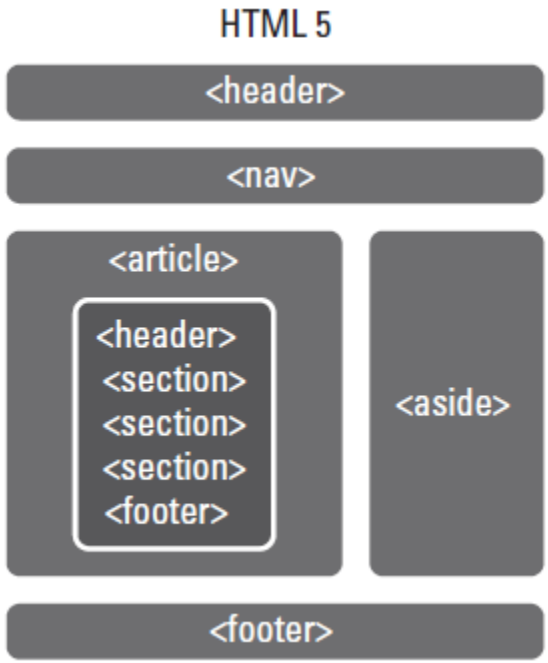
- Tag names that are intuitive
- Makes it easier to build and modify HTML documents
- Makes it easier for Web browsers and other programs to interpret
- Developers can still use `<div>` in HTML5 documents; should use new structure elements whenever appropriate

`header` and `footer` Elements

- The `header` element defines a header for a document, section, or article. HTML 4.01 uses the header div ( ).
- The `footer` element defines a footer for a document or section, and typically contains information about the document or section, such as the author name, copyright data, links to related documents, and so on.

header and footer Elements

- Can include multiple headers or footers in an HTML5 document



header and footer Markup Example

```
<article>
  <header>
    <h1>Learning HTML5</h1>
    <h2>The New Elements</h2>
  </header>
  <p>New HTML5 tags make Web page and application
  development easier than ever.</p>
  <footer>
    <p>Published: <time datetime="2012-09-
    03"September 3, 2012</time></p>
  </footer>
</article>
```

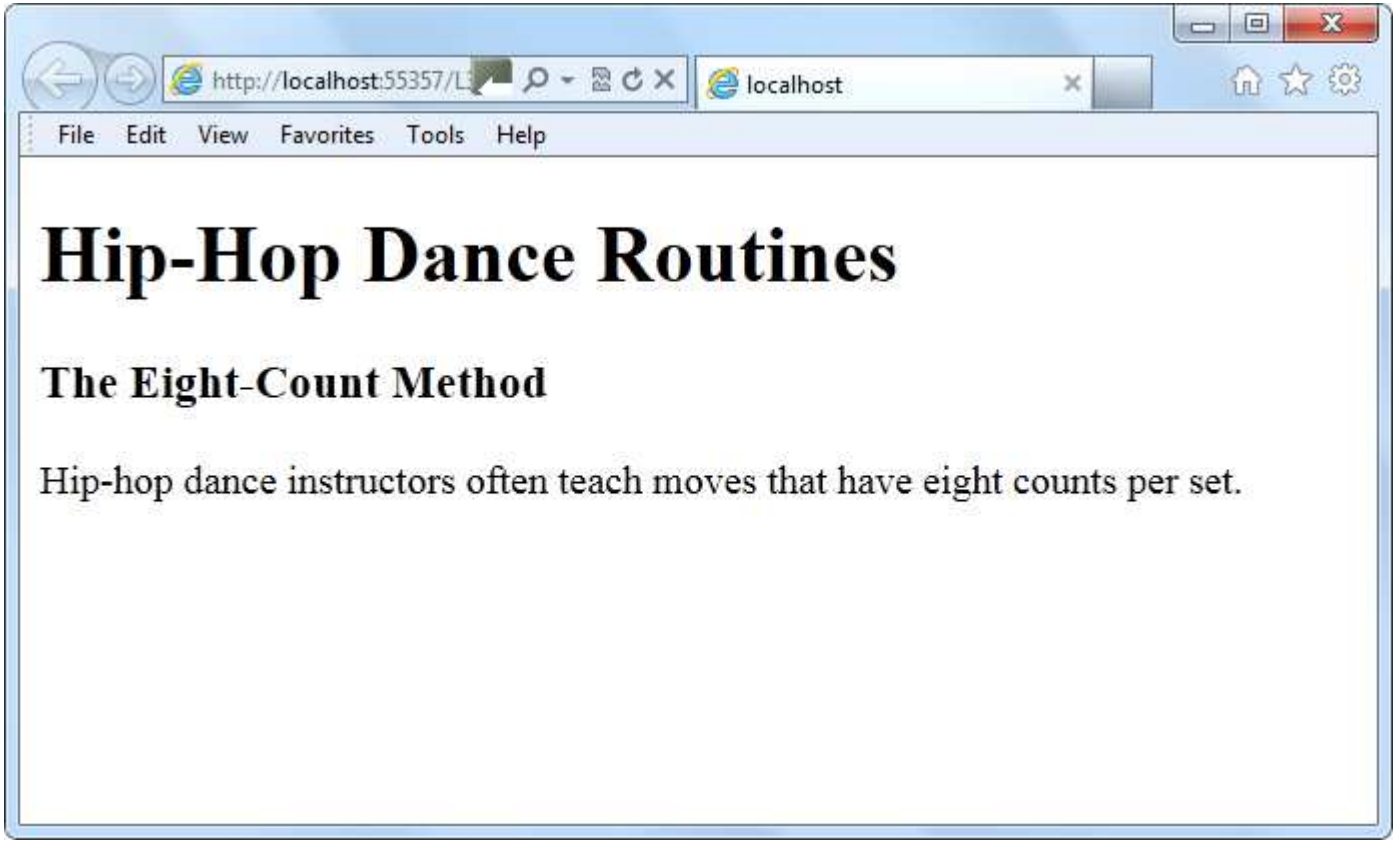
`section` Element

- Defines a section in a document, such as a chapter, parts of a thesis, or parts of a Web page whose content is distinct from each other
- According to the W3C, must contain at least one heading and define something that would ordinarily appear in the document's outline

section Example

```
<section>
    <hgroup>
        <h1>Hip-Hop Dance Routines</h1>
          <h3>The Eight-Count Method</h3>
    </hgroup>
    <article>
      <p>Hip-hop dance instructors often teach
      moves that have eight counts per set.</p>
    </article>
</section>
```

`section` Example



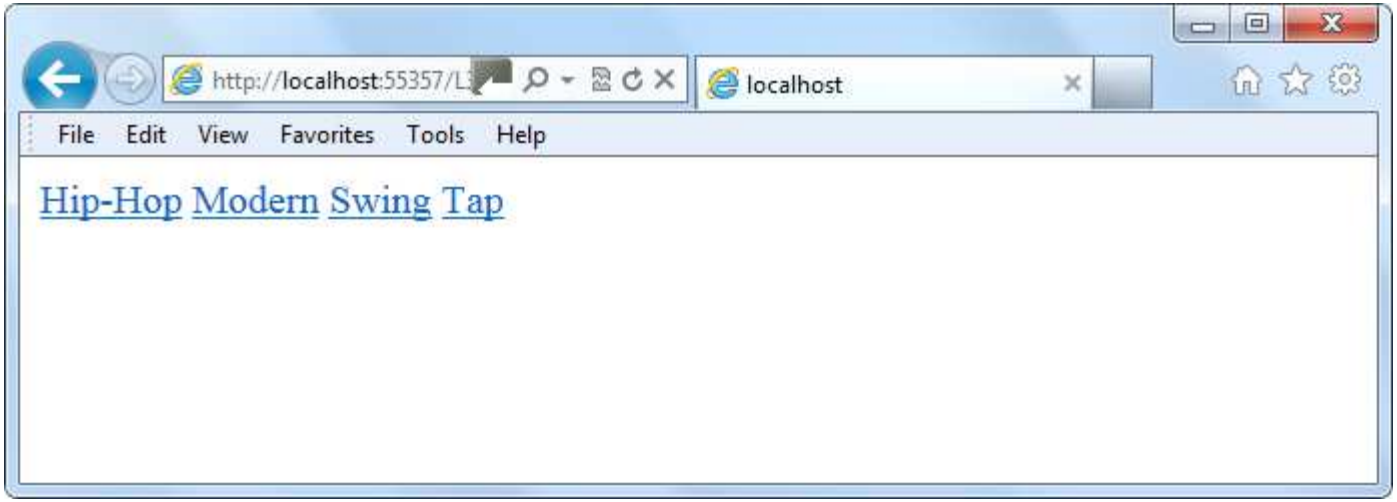
nav Element

- Defines a block of navigation links and is useful for creating
  - A set of navigation links as a document's primary navigation
  - A table of contents
  - Breadcrumbs in a footer
  - Previous-Home-Next links

nav Example

```
<nav>
   <a href="/hiphop/">Hip-Hop</a>
   <a href="/modern/">Modern</a>
   <a href="/swing/">Swing</a>
   <a href="/tap/">Tap</a>
</nav>
```

nav Example


http://localhost:55357/L
localhost
File Edit View Favorites Tools Help
Hip-Hop Modern Swing Tap

`article` Element

- Defines a part of an HTML document that consists of a “self-contained composition” independent from the rest of the content in the document
- Content set off by `<article>` tags can be distributed in syndication
  - Think of it as content that makes sense on its own

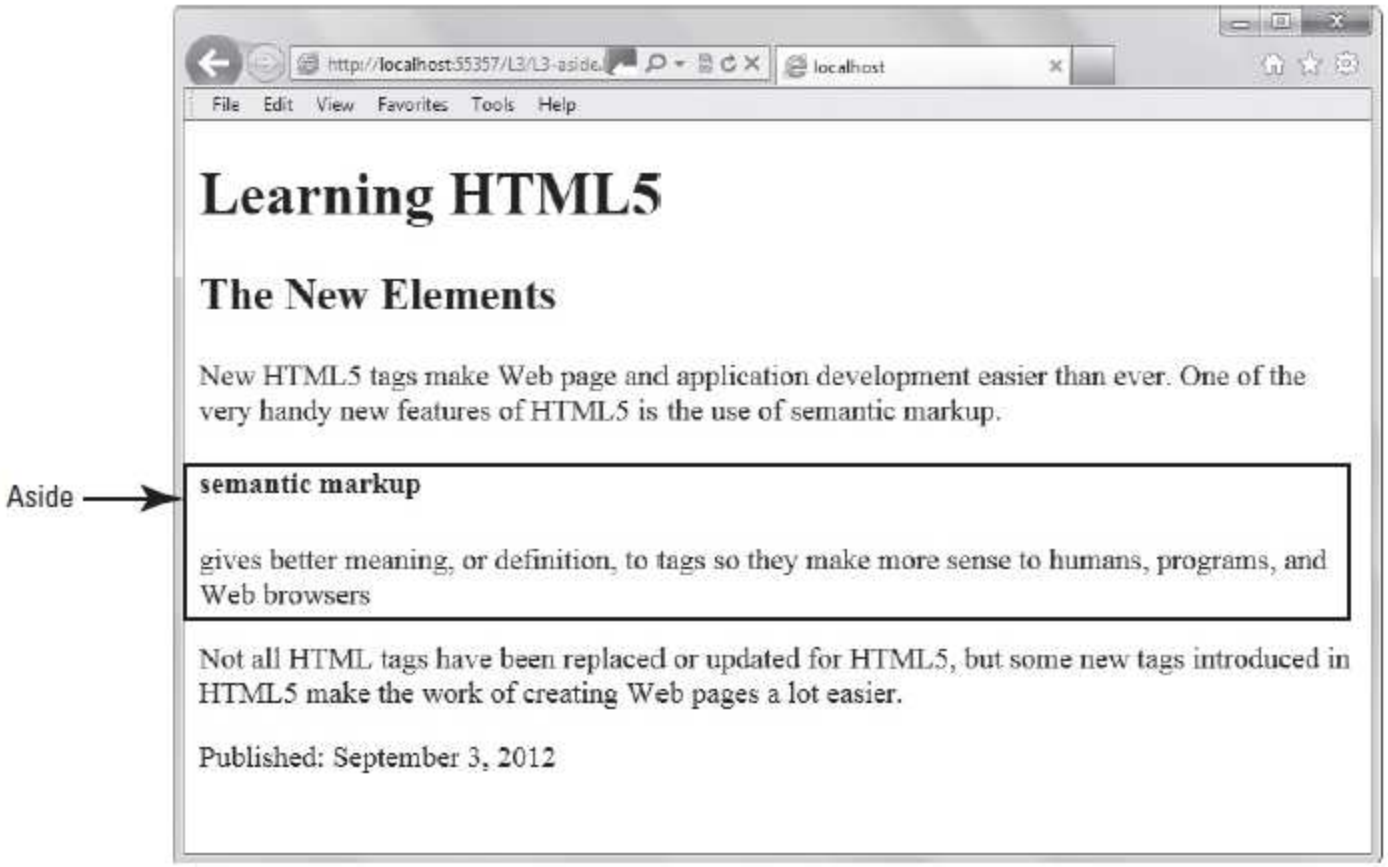
`aside` Element

- Used for sidebars and notes—content that's related to the current topic but would interrupt the flow of the document if left inline

aside Example

```
<article>
  <header>
    <h1>Learning HTML5</h1>
    <h2>The New Elements</h2>
  </header>
  <p>New HTML5 tags make Web page and application
  development easier than ever. One of the very
  handy new features of HTML5 is the use of
  semantic markup.</p>
  <aside>
      <h4><b>semantic markup</b></h4>
        <p> gives better meaning, or definition,
        to tags so they make more sense to humans,
        programs, and Web browsers</p>
    </aside>
  <p>Not all HTML tags have been replaced or updated
  for HTML5, but some new tags introduced in HTML5
  make the work of creating Web pages a lot
  easier.</p>
<footer>
    <p>Published: <time datetime="2012-09-
    03">September 3, 2012</time></p>
  </footer>
</article>
```

aside Example



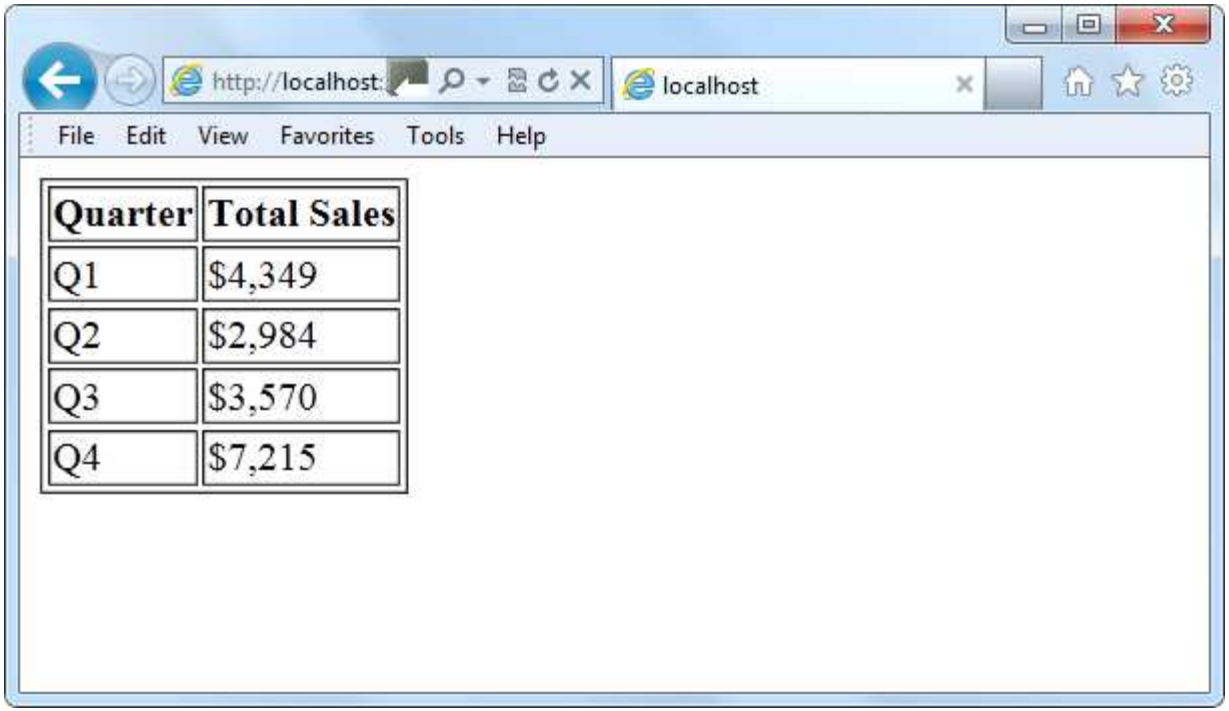
HTML Tables

- `<table>` defines overall table
- `<tr>` defines rows
- `<th>` defines column headers
- `<td>` defines cells
- `<caption>` adds a caption above or below table
- `<col>` applies inline CSS styles
- Long, scrolling tables use `<thead>`, `<tfoot>`, and `<tbody>` tags

HTML Table Example

```
<table border="1">
  <tr> <!--first row-->
    <th>Quarter</th> <!--first column in first row-->
    <th>Total Sales</th> <!--first row, second column-->
  </tr>
  <tr> <!--second row-->
    <td>Q1</td>
    <td>$4,349</td>
  </tr>


  <tr> <!--third row-->
    <td>Q2</td>
    <td>$2,984</td>
  </tr>
  <tr> <!--fourth row-->
    <td>Q3</td>
    <td>$3,570</td>
  </tr>
  <tr> <!--fifth row-->
    <td>Q4</td>
    <td>$7,215</td>
  </tr>
</table>
```

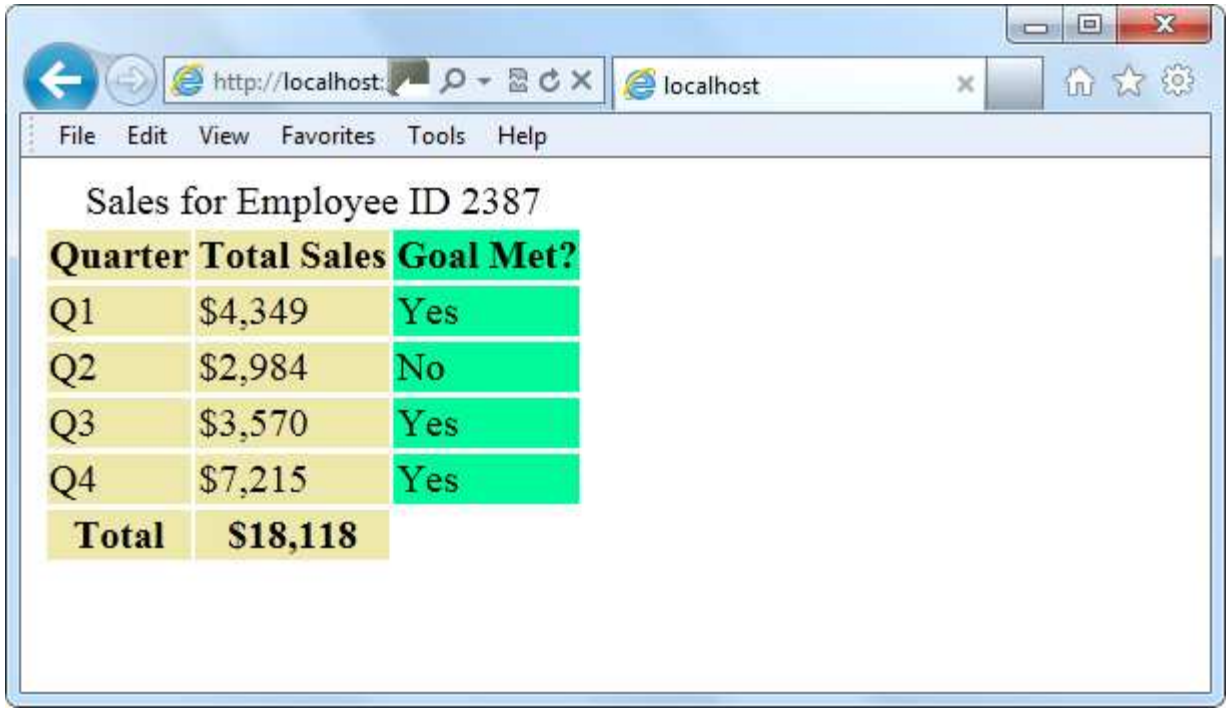
HTML Table Example



HTML Table Example with Color

- When creating a long table that requires scrolling within a browser, use the <thead>, <tfoot>, and <tbody> tags.
  - The content within the table header and footer will remain on the page while the content marked by <tbody> will scroll between them.
- The <thead> tag creates column headings (bolded by default), and the <tfoot> tag is used to display the last row, such as a totals row.
- The <tbody> tag defines all of the content between the header and footer.

HTML Table Example with Color



HTML

- **Links**
  - Any object such as text, graphic images etc. that leads us to a new page on.
- **Text links**
  - Creating a text link use <A> </A> tag
    - <A HREF="http://www.yahoo.com">Click here to visit Yahoo</A>
- **Image links**
  - Replace link text between <A> and </A> with an <IMG> tag that displays an image file.

HTML

- **Email links**
  - Links that when you click on them your email program starts a "compose new message" window that its receiver address is entered from web page
  - This email address is the address you want email to be sent to.

<A HREF="mailto:webmaster@usiu.ac.ke"> Email Me</A>

  - If you want, you can use a subject for the email.

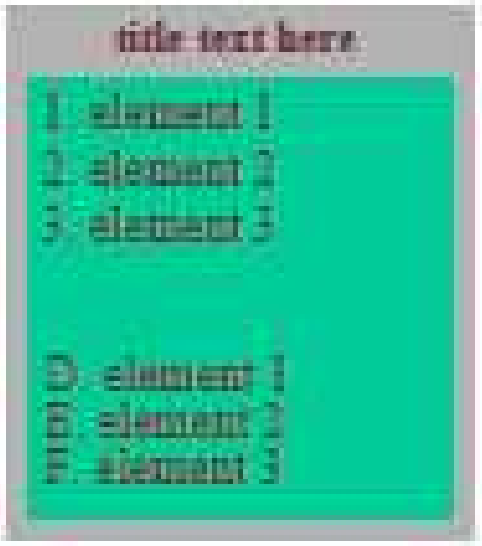
<A HREF="mailto:webmaster@usiu.ac.ke? subject: comments about your site">Email Me</A>

HTML

- **Lists**
  - There are times that you want to insert items related to a subject in list form in your web page.
  - HTML provides you with <UL></UL> tags to do this.
  - If you want the items to start with numbers instead of bullets, you must use <OL></OL> tags instead of <UL></UL> tags.
- **Horizontal Separator Rule**
  - Another useful html tag that you will use, is <HR> tag.
  - If you need to separate text in your web page by horizontal lines , you use <HR> tag.

Some HTML Tags

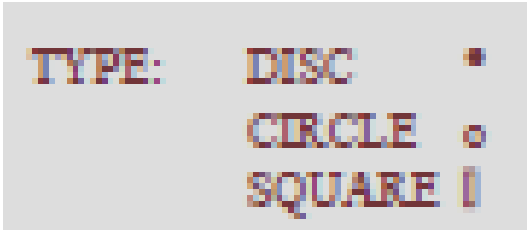
- **Ordered lists:**
- **<OL>**
  - **<LI>**element 1</LI>
  - **<LI>**element 2</LI>
  - **<LI>**element 3</LI>
- **</OL>**
- **<OL TYPE=A START=4 COMPACT>**
  - **<LI>**element 1</LI>
  - **<LI>**element 2</LI>
  - **<LI>**element 3</LI>
- **</OL>**



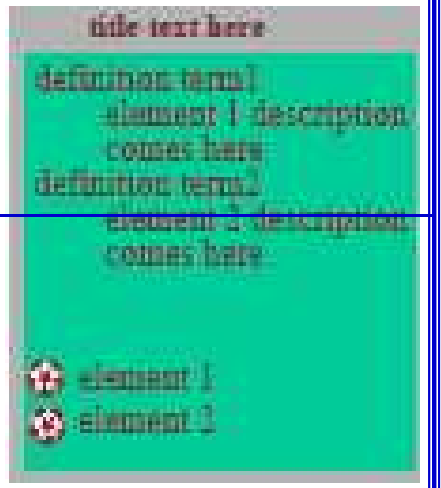
TYPE:		
	1	Arabic numbers
	A	Alphabetic uppercase
	a	Alphabetic lowercase
	I	Roman numeral uppercase
	i	Roman numeral lowercase

Some HTML Tags

- **Unordered lists:**
- **<UL>**
  - **<LI>**element 1</LI>
  - **<LI>**element 2</LI>
  - **<LI>**element 3</LI>
- **</UL>**
  - **<UL TYPE=SQUARE>**
  - **<LI>**element 1</LI>
  - **<LI>**element 2</LI>
  - **<LI>**element 3</LI>
- **</UL>**



Some HTML Tags



- **Definition lists:**
- **<DL COMPACT>**
  - **<DT>**definition term1 **</DT>**
  - **<DD>**element 1 description comes here **</DD>**
  - **<DT>**definition term2 **</DT>**
  - **<DD>**element 2 description comes here **</DD>**
- **</DL>**
- **Definition lists with image bullets:**
  - **<DL>**
  - **<DT><IMG SRC="soccer.gif" ALT="*">** element 1
  - **<DT><IMG SRC="soccer.gif" ALT="*">** element 2
- **</DL>**

HTML

- **Horizontal Separator Rule**
  - <HR WIDTH="50%"> width in percent
  - <HR WIDTH="100"> width in pixels
  - You can also determine line size parameter to change line diameter. <HR size=5>
  - Horizontal rule created by this tag is a line that has a shade. You can change the rule to a solid line instead of a shaded line, you can add a NOSHADE parameter.
    - <HR SIZE=1 NOSHADE>
  - You can also determine Color for your line:
    - <HR color="#000000">

HTML

- **Drawing a table**
  - To draw a table we will use <TABLE> tag.
  - We will need two other related tags to make table rows and columns.
    - These are <TR> and <TD> tags.
    - <TR> tag is used to create a row in table.
    - Data that will fit in a row will be enclosed in <TR> </TR> tags.
    - Use <TD> tag to create columns
  - If you want a cell to be empty, you cannot omit definition for that cell. Insert cell definition, and enter a   between <TD></TD> tags
- **Specifying table sizes**
  - You can specify width for a table both in percents of page width and in pixels.

HTML

- **Text alignment in table cells**
  - By default text entered in a cell will appear at the left side of the cell. You can add either of these options to <TD> tag to specify horizontal alignment of text.
    - <TD ALIGN=CENTER> or
    - <TD ALIGN=RIGHT> or
    - <TD ALIGN=LEFT>
  - Left alignment is default for cells.
  - You can also determine vertical alignment of text in a cell by adding VALIGN option to <TD> tag.
    - There are three values for VALIGN option : TOP, BOTTOM and MIDDLE.
    - MIDDLE is default value if you do not use this parameter.

HTML

- **Images in table cells**
  - You can insert an image in a table cell by inserting <IMG> tag between <TD></TD> tags of a certain cell.
- **Cell Width (Column Width)**
  - You can determine width of each column in your table by specifying width of cells in first row.

```
<TABLE WIDTH=400 HEIGHT=100 BORDER=3>
<TR>
<TD WIDTH=140>TOP LEFT</TD>
<TD WIDTH=260>TOP RIGHT</TD>
</TR>
```

  - You can also determine cell widths in percent. Sum of cell width percentages must be 100%.

```
<TR>
<TD WIDTH=35%>TOP LEFT</TD>
<TD WIDTH=65%>TOP RIGHT</TD>
</TR>
```

HTML

- **Cell padding**
  - Cell padding is the space between cell borders and table contents such as text, image etc.

    <TABLE BORDER=3 CELLPADDING=20>
- **Cell spacing**
  - Cell spacing parameter determines the space between inner and outer parts of a table.
  - If you increase this value you will have a thick border.
  - Default value for this property is 2.
  - If you specify 0 for it, you will have a very thin border.

HTML

Column Span

- Sometimes you need to join two cells in a row to each other.
- For example in a 2*3 table we may want to join two cells with each other .
- In this way we will have two cells in first row and three cells in second row.

Row Span

- This time we want to join two cells in a column (from different rows).
- This is the same as previous section with the difference that we will join cells from different rows rather than cells in different columns.
- This time we must use ROWSPAN instead of COLSPAN.

HTML

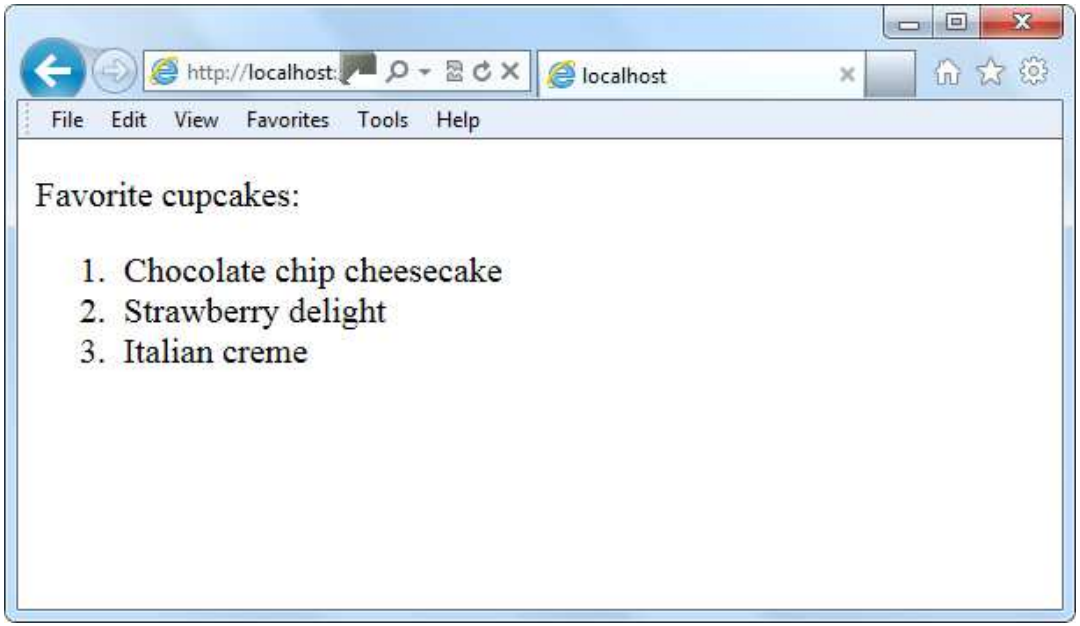
- **Nested Tables**
  - Yes we can nest tables in each other. If you are going to design complicated web pages you will always do this.

Ordered List

- Orders list entries using numbers, by default
- Uses the `<ol>` tag with attributes:
  - `reversed`: Reverses the order of the list
  - start number: Specifies the start value of the ordered list
  - type: Specifies list item marker, such as "1" for displaying decimal numbers

Ordered List Example

```
<p>Favorite cupcakes:</p>
<ol>
  <li>Chocolate chip cheesecake</li>
  <li>Strawberry delight</li>
  <li>Italian creme</li>
</ol>
```

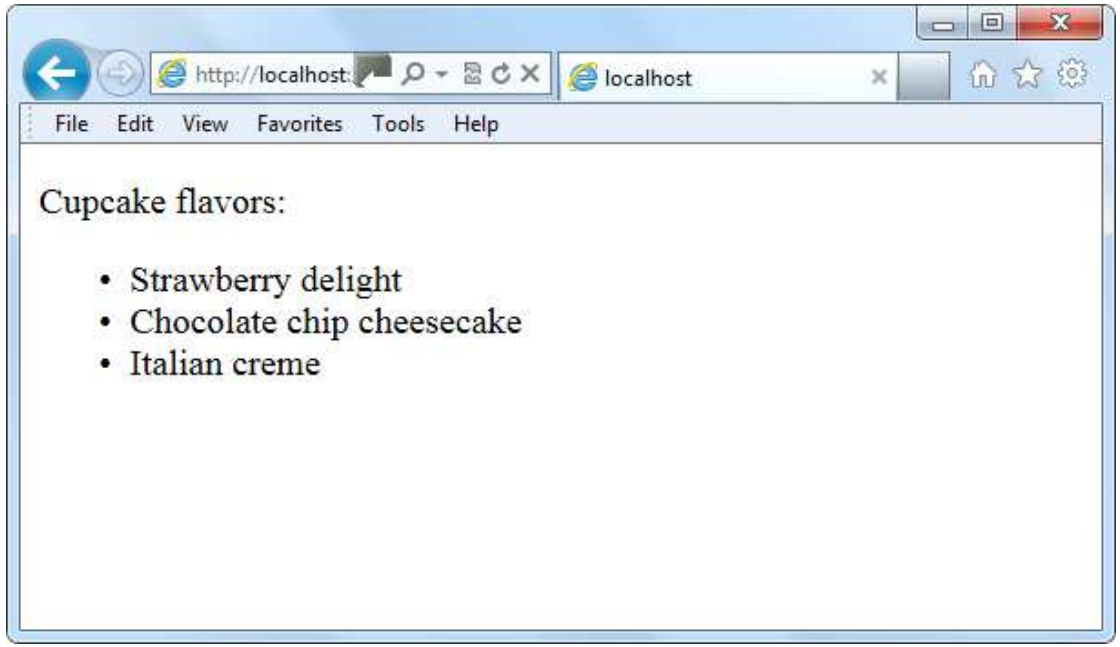


Unordered List

- Displays list entries in a bulleted list
- Uses a `<ul>` tag
- Round bullet symbol is the default marker for list items
- Can change bullet symbols
  - For squares, add `type="square"` to the `<ul>` tag
  - For empty circles, add `type="circle"`

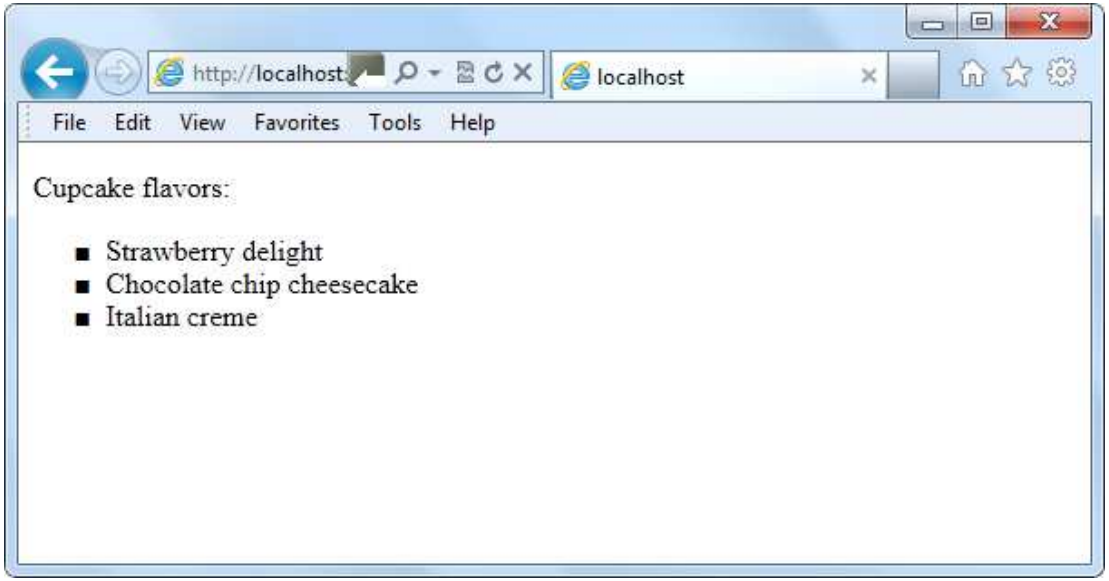
Unordered List Example

```
<p>Cupcake flavors:</p>
<ul>
  <li>Strawberry delight</li>
  <li>Chocolate chip cheesecake</li>
  <li>Italian creme</li>
</ul>
```



Unordered List Example

```
<p>Cupcake flavors:</p>
<ul type="square">
  <li>Strawberry delight</li>
  <li>Chocolate chip cheesecake</li>
  <li>Italian creme</li>
</ul>
```



Understanding How Forms Work

- The data-processing software can then work with the data and send a response back to the user
- The user enters data via an HTML form

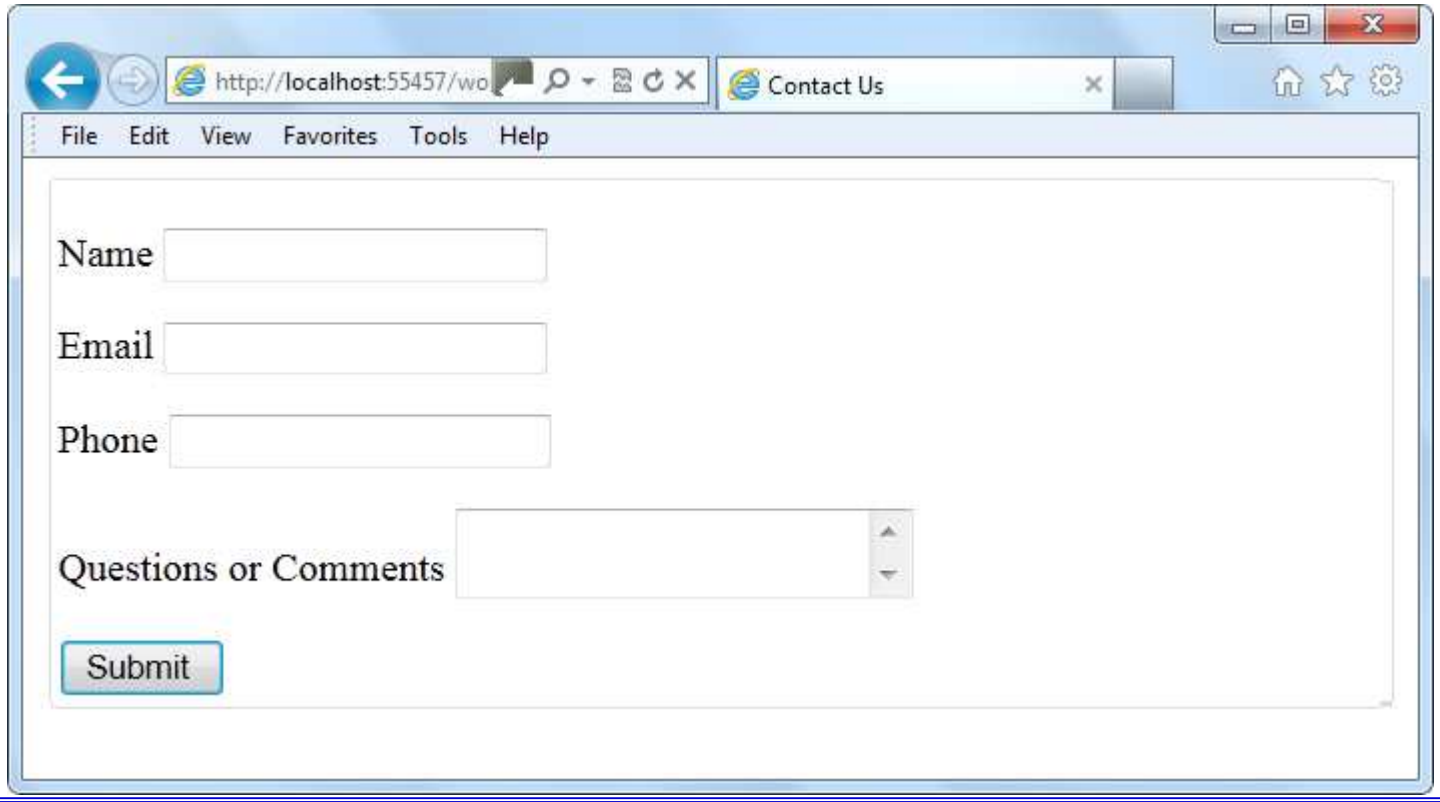
Creating a Form

- Use the `<form>` start and end tags
- All form content and fields are between `<form>` tags
- Common syntax:

```
<form id="keyword">
<content and fields>
</form>
```

Creating a Form (Continued)

- The `fieldset` element is used with many forms to group related elements.
- The `<fieldset>` tag draws a box around individual elements and/or around the entire form.



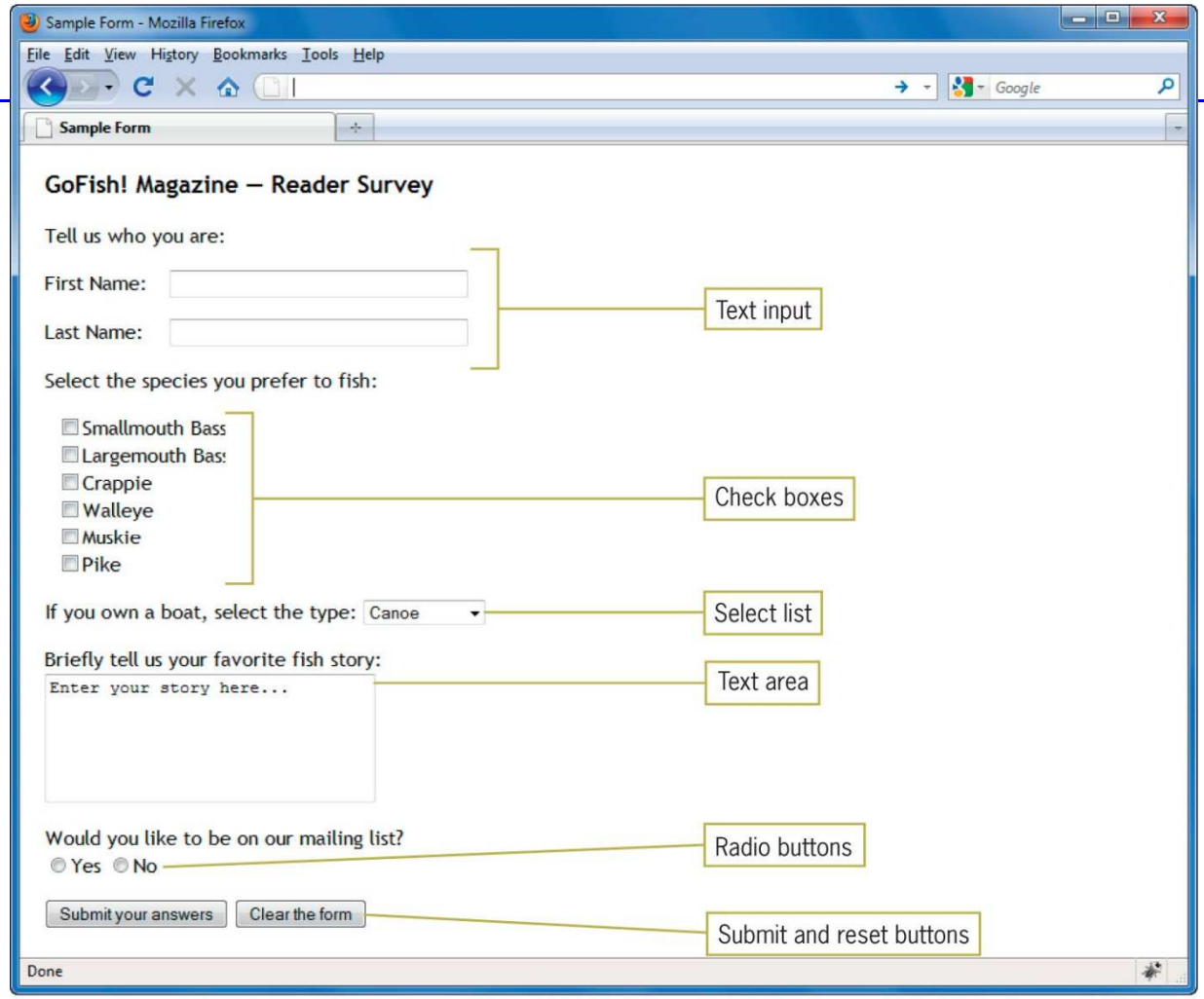


Figure 11-1 Sample HTML form

Using the <form> element

- The <form> element is the container for creating a form
- A variety of attributes describe how the form data will be processed

Attribute	Description
action	The URL of the application that processes the form data; this URL points to a script file or an e-mail address
enctype	The content type used to submit the form to the server (when the value of the method is "post"); most forms do not need this attribute
method	Specifies the HTTP method used to submit the form data; the default value is "get" • get—The form data is appended to the URL specified in the action attribute • post—The form data is sent to the server as a separate message
accept	A comma-separated list of content types that a server processing this form can handle correctly; most forms do not need this attribute
accept-charset	A list of allowed character sets for input data that is accepted by the server processing this form; most forms do not need this attribute

Table 11-1 Form Attributes

Using the <form> element

- The following code shows a typical <form> element:

```
<form method="post"
action="https://signup.website.com/register.asp">
```

Using get or post

- The difference between *get* and *post* is the way the data is sent to the server
- method="get": this method sends the form information by including it in the URL
- method="post": this method sends the form information securely to the server within the message body

Using the mailto Action

- Lets you collect data from a form and send it to any e-mail address

```
<form action="mailto:joel@joelsklar.com"
method="post" enctype="text/plain">
```

Creating Input Objects

- The <input> element defines many of the form input object types
- The type attribute specifies the type of input object

Type Attribute Value	Description
text	Creates a text entry field that lets the user enter a single word or a line of text; this is the default object type
password	Creates the same type of text entry field created by the value "text," but the user entry is masked by asterisks
checkbox	Provides on/off toggles that the user selects; check boxes are best used with multiple-answer questions, and multiple check boxes can contain the same name, letting you group them together so that users can select multiple values for the same property.
radio	Lets a user choose one value from a range of values; when radio buttons are grouped together with the same name, only one choice can be selected
submit	Sends the form data to the server using the transmission method specified in the <form> element; every form needs a submit button
reset	Clears the form of any user-entered data and returns it to its original state
hidden	Adds a control that is not displayed in the browser; the hidden type is useful for sending additional information with the form data that may be needed for processing
image	Adds a graphic button to the form, rather than the default button
button	Creates a button that has no default behavior; the button's function is usually defined by a script; when the user pushes the button, the script function is triggered
file	Lets the user select a file that is submitted with the form

Table 11-2 <input> Element Types

Labeling Form Elements

- The <label> element lets you create a caption for an input element
- Lets you extend the clickable area of a form element

```
<p>
<label class="username" >First Name:</label>
<input type="text" name="firstname"
size="35" maxlength="35" />
</p>
```

Labeling Form Elements

- To make the text clickable, you associate the <label> element with the <input> element by using the *for* and *id* attributes

```
<p>
<label class="username" for="First Name">
First Name:</label>
<input type="text" name="fi rstname" id="First
Name"
size="35" maxlength="35" />
</p>
```

Creating Text Boxes

- The text box is the most commonly used form element

```
<input type="text" name="firstname"
size="20" maxlength="35" value="First
Name">
```

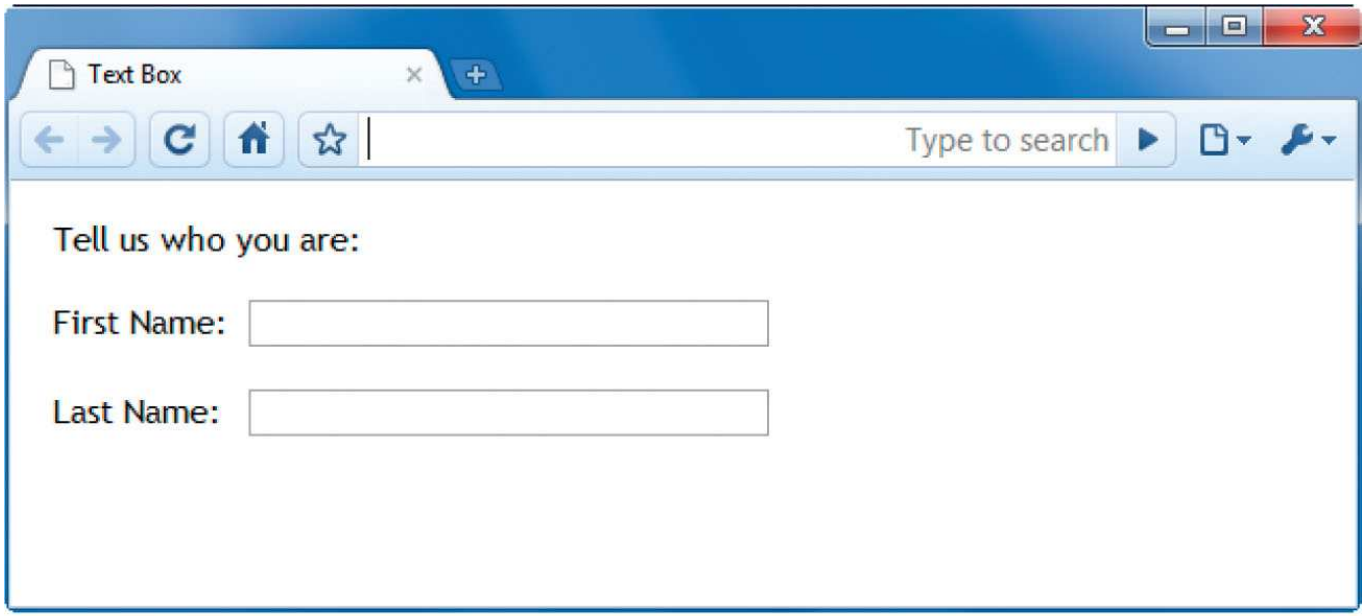


Figure 11-4 Text box input type

Creating Check Boxes

- Check boxes are an on/off toggle that the user can select

```
<input type="checkbox" name="species"
value="smbass"> Smallmouth Bass
```

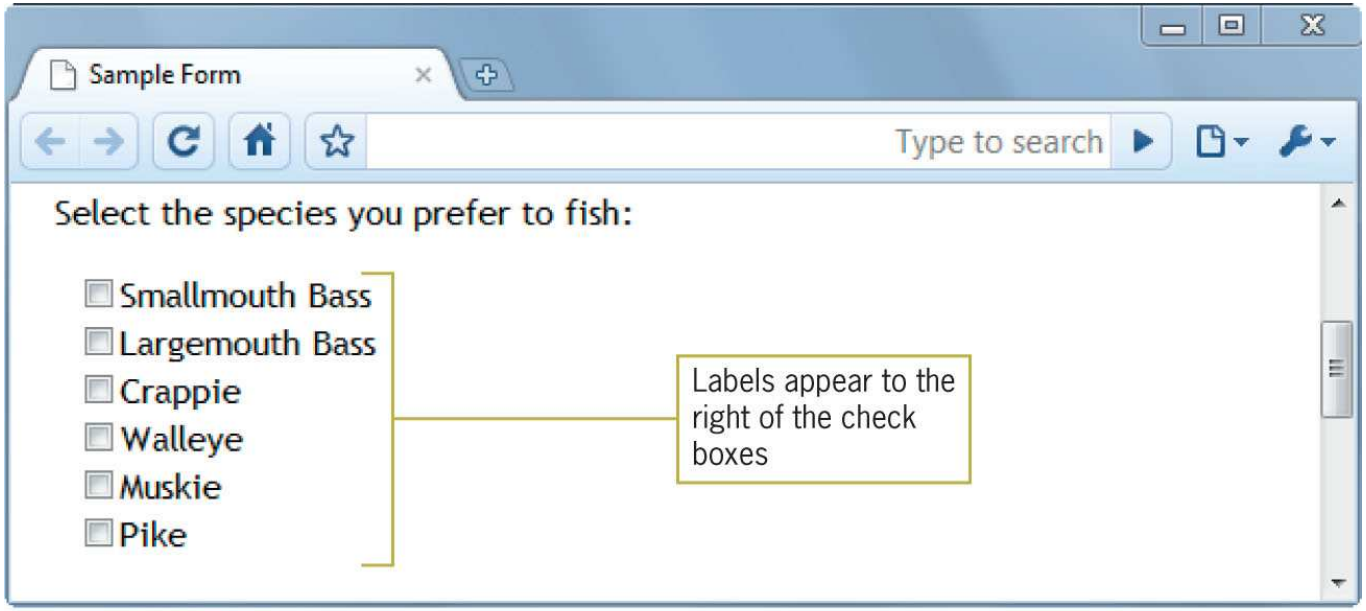


Figure 11-5 Check box input type

Creating Radio Buttons

- Radio buttons are like check boxes, but only one selection is allowed

```
<p>Would you like to be on our mailing list?</p>
<p><input type="radio" name="list" value="yes"
id="Yes" />
<label for="Yes">Yes</label>
<input type="radio" name="list" value="no"
id="No" />
<label for="No">No</label>
</p>
```

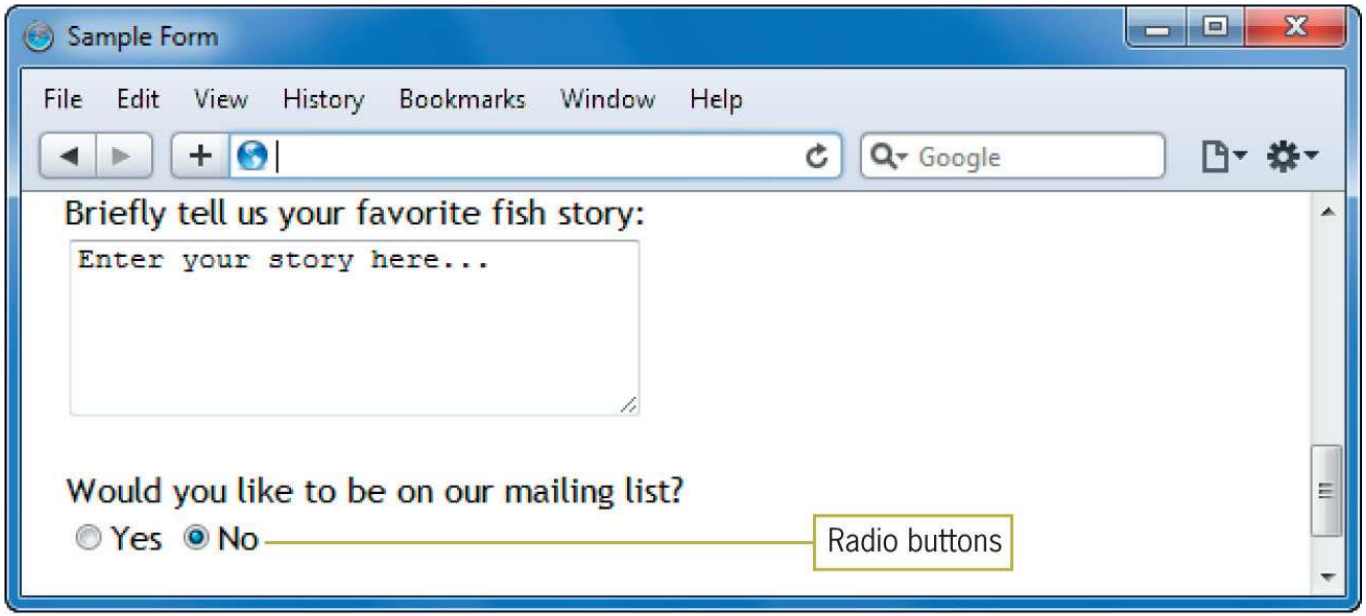


Figure 11-6 Radio buttons input type

Creating Submit & Reset Buttons

- The submit and reset buttons let the user choose whether to send the form data or start over

```
<input type="submit" value="Submit your
answers">
<input type="reset" value="Clear the
form">
```

Figure 11-7 Submit and reset input buttons

Creating an Image for the Submit Button

- You can choose an image file and use it instead of the default submit button

```
<input type="image" src="submit.gif"
alt="submit button">
```

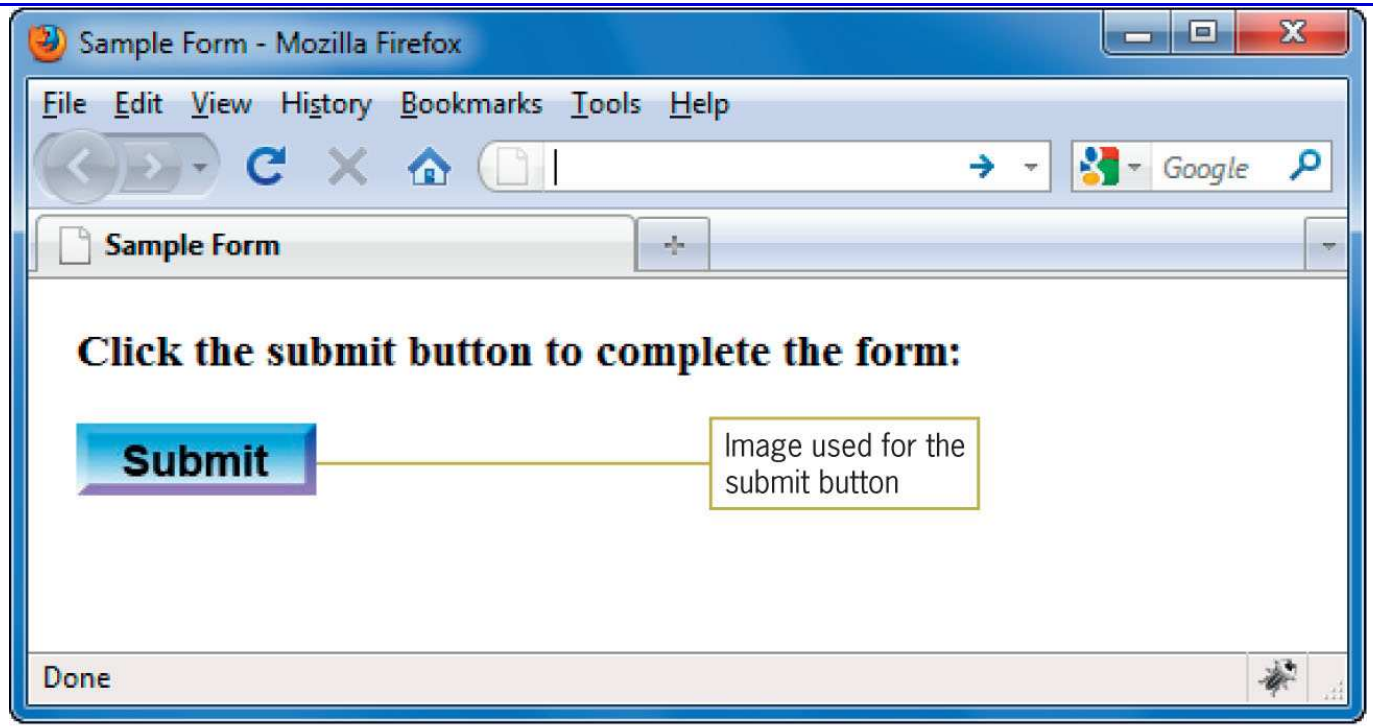


Figure 11-8 Using an image for the submit button

Letting the User Submit a File

- Users can select a file on their own computer and send it to the server

```
<p>Use the browse button to select your file:</p>
<p><input type="file" size="30“></p>
```

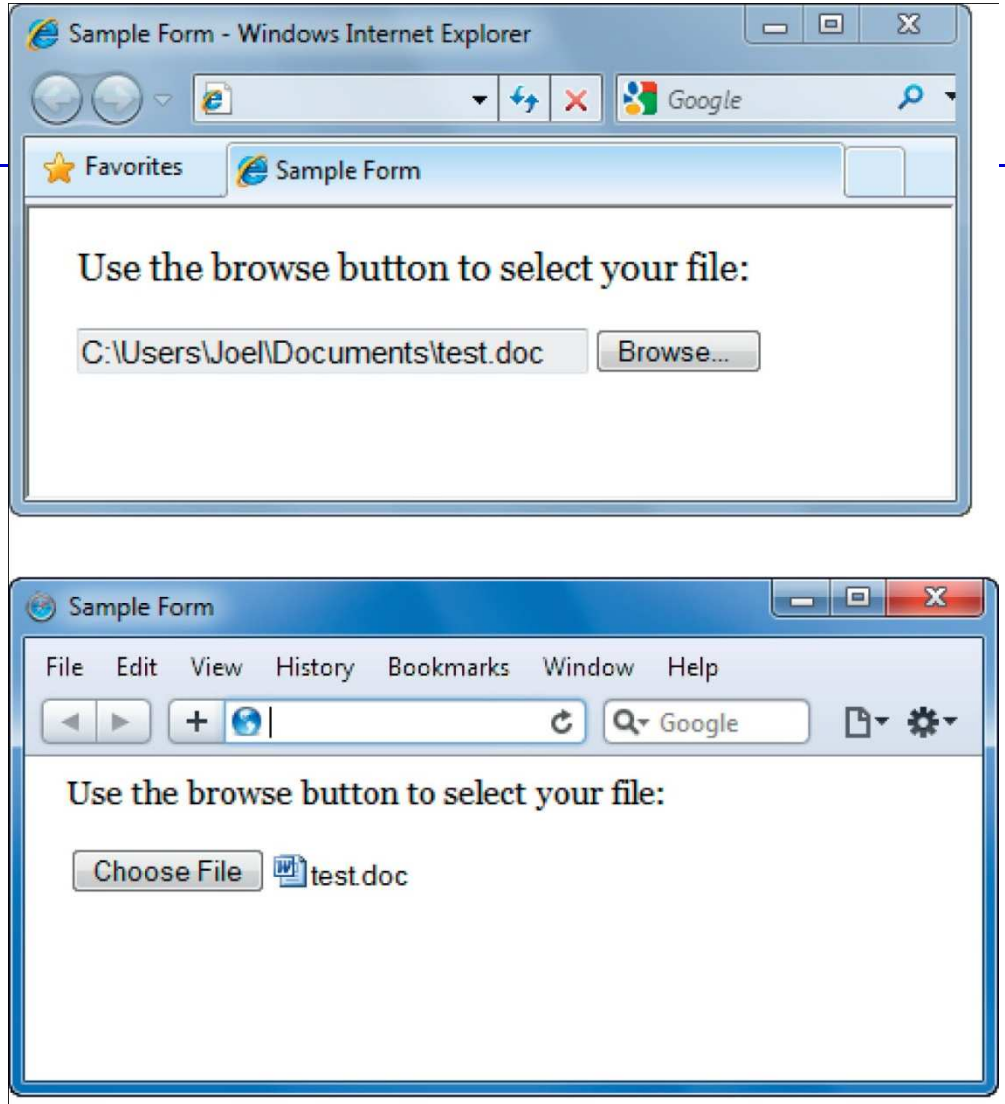


Figure 11-9 File input type in Internet Explorer (top) and Safari (bottom)

Creating a Password Entry Field

- The password input box works like the text input, but the entered text is hidden by asterisks

```
<p>Enter your user name and password:</p>
<p>
User Name: <input type="text" size="30" />
Password: <input type="password" size="30" />
</p>
```

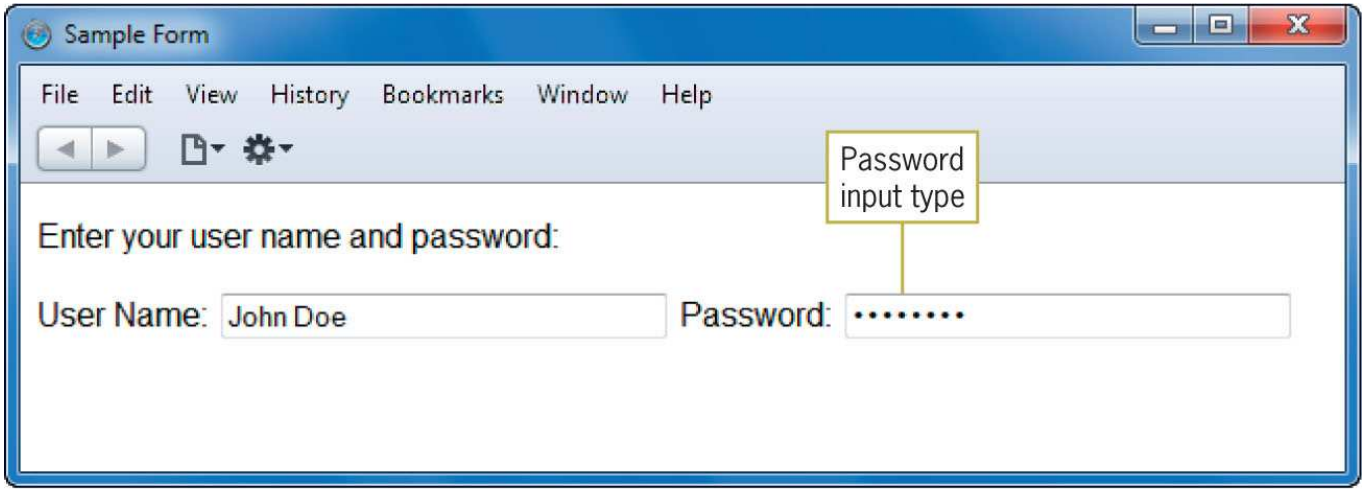


Figure 11-10 Password input type

Using the <select> Element

- The <select> element lets you create a list box or scrollable list of selectable options

```
<select name="boats">
      <option>Canoe</option>
      <option>Jon Boat</option>
      <option>Kayak</option>
      <option>Bass Boat</option>
      <option>Family Boat</option>
</select>
```

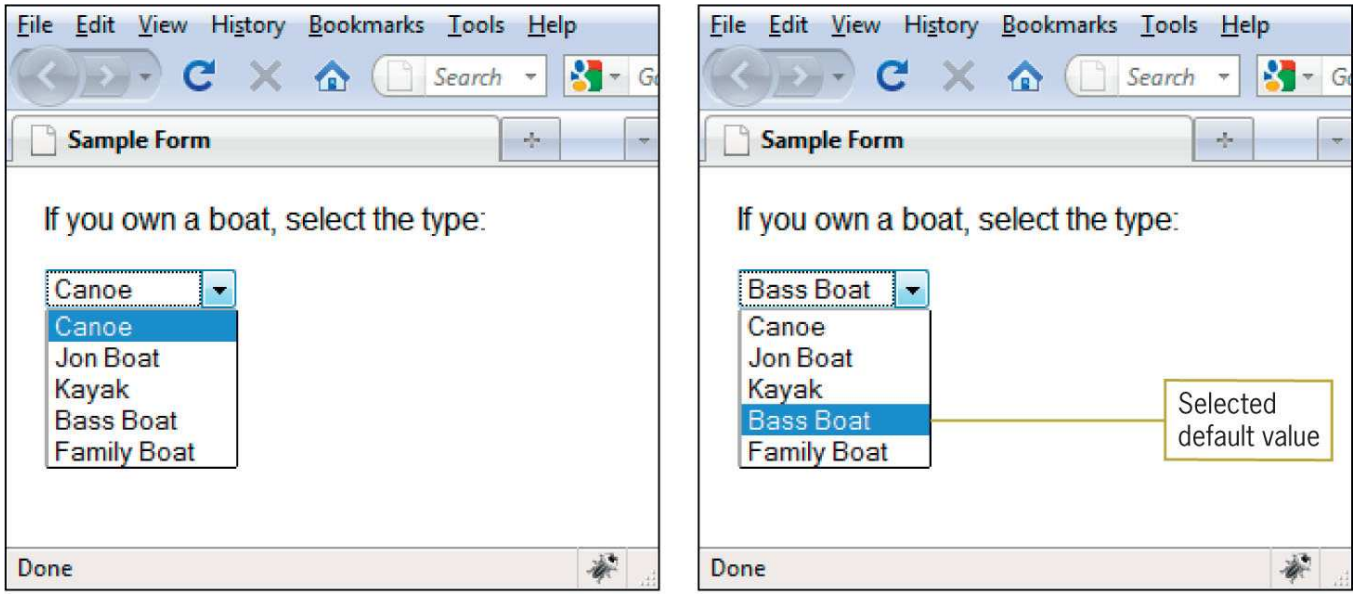


Figure 11-11 Select list element

Using the <select> Element

- You can choose to let the user pick multiple values from the list by adding the multiple attribute

```
<select name="snacks" multiple size="6">
     <option>Potato Chips</option>
     <option>Popcorn</option>
     <option>Peanuts</option>
     <option>Pretzels</option>
     <option>Nachos</option>
     <option>Pizza</option>
     <option>Fries</option>
</select>
```

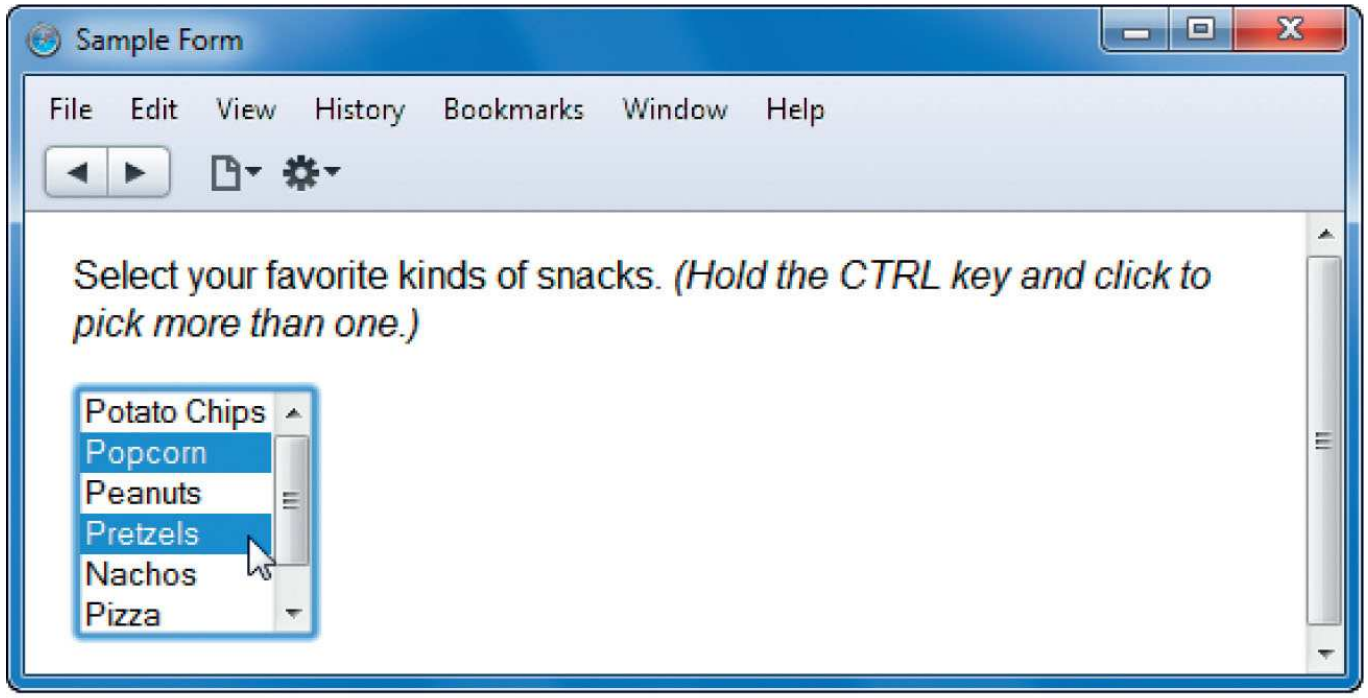


Figure 11-12 Scrollable select list with multiple choices

Using the <select> Element

- You group and label sets of list options with the <optgroup> element and label attribute

```
<optgroup label="Salty Snacks">
      <option>Potato Chips</option>
      <option>Popcorn</option>
      <option>Peanuts</option>
      <option>Pretzels</option>
</optgroup>
```

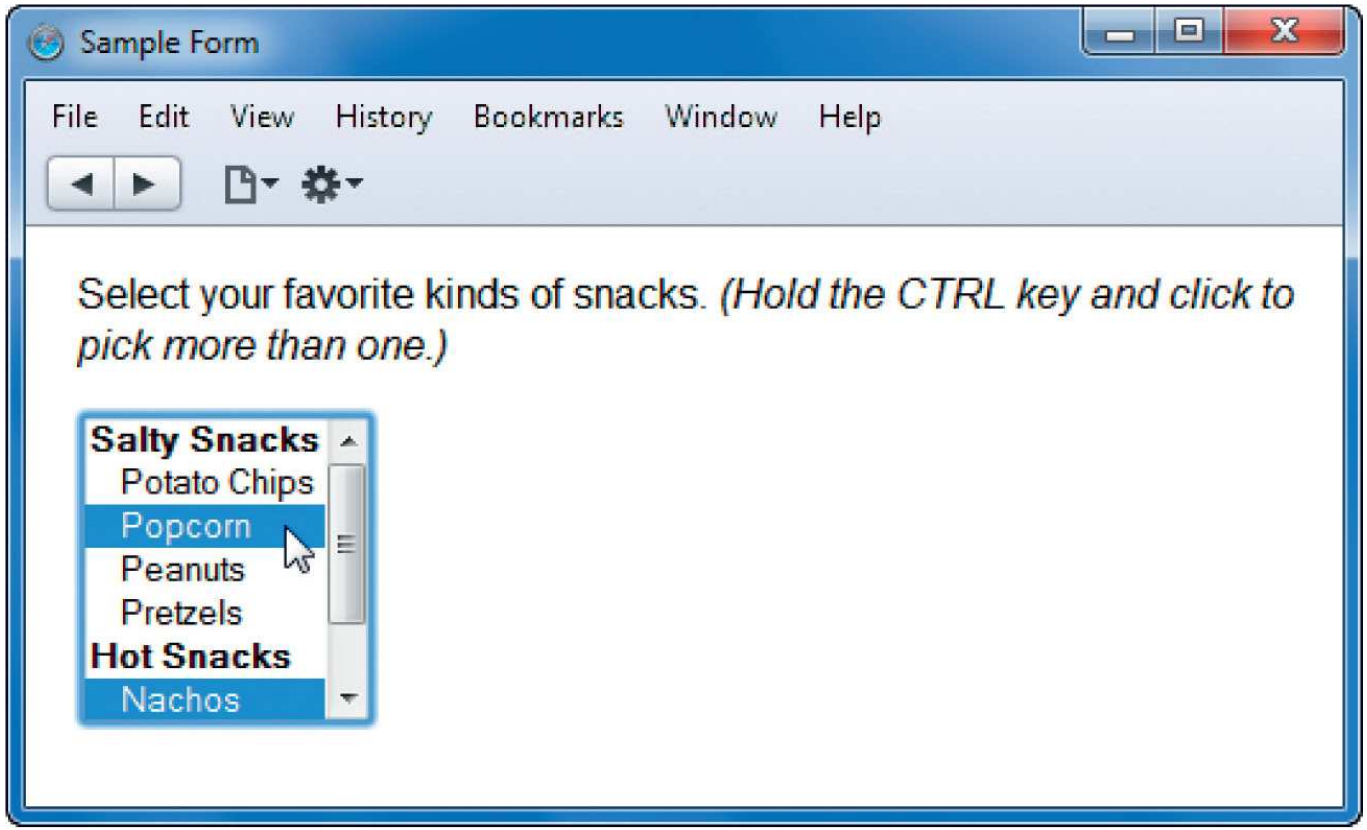


Figure 11-13 Grouping list options

Using the <textarea> Element

- The <textarea> element lets you create a larger text area for user input

```
<p><b>Briefly tell us your favorite fish
story:</b><br>
<textarea name="fishstory" rows="5"
cols="30">
Enter your story here...
</textarea>
</p>
```

Figure 11-14 Using the <textarea> element

Creating Input Groupings

- You can use the <fieldset> and <legend> elements to create groupings of different types of input elements

```
<fieldset>
<legend><b>Select the species you prefer
to fish:</b></legend>
<input type="checkbox" name="species"
value="smbass"> Smallmouth Bass
<input type="checkbox" name="species"
value="lgbass"> Largemouth Bass <br>
<input type="checkbox" name="species"
value="pike"> Pike
</fieldset>
```

HTML5 Form Semantics: New Form Elements

- Form input is the information a user enters into fields in a Web or client application form.
- HTML5 introduces several new form and input element attributes; some are:
  - `url` for entering a single Web address
  - `email` for a single email address or a list of email addresses
  - `search` to prompt users to enter text they want to search for

Form `required` and `email` Attributes

- The ***`required` attribute*** requires information in a field when the form is submitted.
- The ***`email` attribute*** requires the user to enter an email address.
- Markup example:

```
<input type="email" required />
```

required Example

Form `placeholder` Attribute

- **Placeholder text** is text displayed inside an input field when the field is empty. It helps users understand the type of information they should enter or select. When you click on or tab to the input field and start typing, the newly entered text replaces the placeholder text.
- Markup example:

```
<input name="fName" placeholder="First
Name" />
```

Form `pattern` Attribute

- The ***`pattern` attribute*** provides a format (a regular expression) for an input field, which is used to validate whatever is entered into the field.
- Markup example:

```
<input type="text" id="empID"
name="EmployeeID" required
pattern="[A-Z]{2}[0-9]{4}"
title="Employee ID is two capital
letters followed by four digits">
```

Form `pattern` Attribute

- You can use the `pattern` attribute with these `<input>` types:
  - `text`
  - `search`
  - `url`
  - `telephone`
  - `email`
  - `password`

Form `autofocus` Attribute

- The ***`autofocus` attribute*** moves the focus to a particular input field when a Web page loads.
- Markup example:

```
<input type="text" name="fname"
autofocus="autofocus" />
```

Form with `datalist` Example

- **<datalist>** : The datalist element is used to dene a predened list of option for a specific input.
- The datalist element is hooked up to an input element using the list attribute on the input element.

```
<form action=“ ” method="post">
Countries : <input list="list1" name="countries1">
<datalist id="list1">
<option value=“Kenya"> Kenya</option>
<option value=“Uganda"> Uganda </option>
<option value=”Sudan"> Sudan </option>
<option value=“Tanzania" > Tanzania </option>
<option value=”Congo"> Congo </option>
</datalist>
<input type="submit" value="Submit" />
</form>
```

Form with color chooser Example

- HTML5 allows new input types for forms. These input types add features for better control and validation.
- The new input types are:
  - **color –** If a browser supports this input type, the intention is that clicking in the text field will result in a color chooser popping up
- **Syntax:** <input type=”color” name=”name” />
- **Attributes :**
  - **Value:** It defines the initial value. This color will be shown in the texteld and also when the color chooser opens it selects that color only by default.

Enter a Color: <input type="color" name="color" value="#FF0000" />

<input type="submit" value="Submit" />

Form with color chooser Example

- HTML5 allows new input types for forms. These input types add features for better control and validation.
- The new input types are:
  - **color –** If a browser supports this input type, the intention is that clicking in the text field will result in a color chooser popping up
- **Syntax:** <input type=”color” name=”name” />
- **Attributes :**
  - **Value:** It defines the initial value. This color will be shown in the texteld and also when the color chooser opens it selects that color only by default.

Enter a Color: <input type="color" name="color" value="#FF0000" />

<input type="submit" value="Submit" />

Form with datepicker Example

- **date –** This works the same as the datepicker. When you click the texteld a calendar is displayed and allows you to select a date from it.
- **Syntax:** <input type=”date” name=”name” />
- **Attributes :**
  - **Value:** The initial value that is by default displayed in the text field.
  - **Step:** The step size in days. By default it is 1.
  - **Min , Max :** The minimum and maximum date that can be selected.

Enter a Date: <input type="date" name="date" />

<input type="submit" value="Submit" />

Form with datetime Example

- **datetime:** This element allows the user to select date and time.
- **Syntax:** <input type=”datetime” name=”name” />
- **Attributes :**
  - **Value=** Sets the initial value.

Form with Email Example

- **email –** This element allows you to enter a valid email address.
- **Syntax:** <input type=”email” name=”name” />
- **Attributes :**
  - **Value:** It is used to dene an initial value.
  - **List:** Denes the id of datalist to be displayed.

Enter an Email Address: <input type="email" name="name" />

<input type="submit" value="Submit" />

Form with Month Example

- **month –** This element allows the user to select a month and year.
- **Syntax:** <input type=”month” name=”name” />
- **Attributes :**
- **Value:** It defines the initial value.
- Note: First you have to select the month or year field and then click the up or down arrow:

```
<input type="month" name="month" />
<input type="submit" value="Submit" />
```

Form with Number Example

- **number –** This element lets you select a number. It is also known as spinner.
- **Syntax:** <input type=”number” name=”name” />
- **Attributes :**
  - **Value:** It denes the initial value. If this eld is not set the field is defined blank.
  - **Step:** It denes the step size. The deference in the selected number is number displayed after clicking up and down arrow is called step. By default it is 1.
  - **min and max:** The minimum and maximum value that can be selected on clicking the up and down arrows.

Enter a Number: <input type="number" value="5" step="2" name="number"/>

<input type="submit" value="Submit" />

Form with Range Example

- **range –** This element allows the user to select a value between the given range.
- **Syntax:** <input type=”range” name=”name” />
- **Attributes :**
  - **Value:** It denes the initial value. Also it is always halfway between the min and max.
  - **Step:** It change in the number on moving the slider is known as step. It is by default 1.
  - **min, max –** Defines the minimum and maximum value of the range.

Enter a Number in the Range (0 to 10): <input type="range" value="5" step="2" name="range" min="0" max="10"/>

<input type="submit" value="Submit" />

Form with Search Example

- **search –** This element allows you to enter a string that can be used to search anything in the database.
- **Syntax:** <input type=”search” name=”name” />
- **Attributes :**
  - **Value:** Sets the initial value.

Search Google: <input type="search" name="search" />

Form with tel Example

- **tel –** This element allows the user to enter a valid telephone no.
- **Syntax:** <input type=”tel” name=”name” />
- **Attributes :**
  - **Value:** Sets an initial value.
- **time –** This element allows the user to select time.
- **Syntax:** <input type=”time” name=”name” />
- **Attributes :**
  - **Value:** Sets the initial value.

```
Enter A Particular Time: <input type="time" name="time" />
<input type="submit" value="Submit" />
```

Form with URL Example

- **url –** This element allows the user to enter a valid url address.
- **Syntax:** <input type=”url” name=”name” />
- **Attributes :**
  - **Value:** Sets the initial value.

Enter A Particular url: <input type="url" name="url" />

- **week –** This element allows the user to select a week and a year.
- **Syntax:** <input type=”week” name=”name” />
- **Attributes :**
  - **Value:** Sets the initial value.

Enter A Particular Week: <input type="week" name="week" />

<input type="submit" value="Submit" />

Meter Element

- **meter-** "Meter" is a new element in HTML5 which represent value of a known range as a gauge. The keyword here is "known range". That means, you are only allowed to use it when you are clearly aware of its minimum value and maximum value.
- One example is score of rating. I would rate this movie

<meter min="0" max="10" value="8">8 of 10</meter>.

```
Science : <meter min="0" max="100" value="95">95 of 100</meter> <br />
Math : <meter min="0" max="100" value="60">60 of 100</meter><br />
Geography : <meter min="0" max="100" value="20">20 of 100</meter> <br />
History : <meter min="0" max="100" value="50">50 of 100</meter>
```

Progress **Element**

- **Progress-** The new "progress" element appears to be very similar to the "meter" element. It is created to indicate progress of a specific task.
- The progress can be either determinate OR interderminate. Which means, you can use "progress" element to indicate a progress that you do not even know how much more work is to be done yet.

Progress of Task A : <progress value="60“ max="100">60%</progress>

Progress of Task A : 

Validation

- The process of verifying that information entered or captured in a form is in the correct format and usable before sending the data to the server
- Some things verified during validation:
  - Required fields are empty
  - Email addresses are valid
  - Dates are valid
  - Text does not appear in a numeric field or vice versa

Validation (Continued)

- Automatic validation of input means the browser checks the data the user inputs.
  - Also referred to as **client-side validation**
- **Server-side validation** occurs when server validates data received from an input form

Validation Example

```
<input type= "text" name= "zipcode"
  pattern= "[0-9] {5}"
  title= "Five-digit zip code" />
```

Keygen Element

- The purpose of the keygen element is to provide a secure way to authenticate users.
- The keygen element is a key-pair generator.
- When a form is submitted, two keys are generated, one private and one public.
- The private key is stored on the client, and the public key is sent to the server.
- The public key could be used to generate a client certificate to authenticate the user in the future.
- Currently, the browser support for this element is not good enough to be a useful security standard.

Keygen Element

- Example :

```
<form action="demo_form.asp"
  method="get">
  Username: <input type="text"
  name="usr_name" />
  Encryption: <keygen name="security"
  />
  <input type="submit" />
  </form>
```

Exercise

- The web page should include
  - A title “A site’s navigational system”
  - Link page your home page and two future pages using a navigation bar
  - Use graphics as links to visit your three favourite pages at the bottom of the page
  - Have a table showing the courses you are taking this semester
  - Add a new web page that has a form
  - To show people where they are, make the link corresponding to this current page different.
    - This lets users know that if they can’t go to that page, they must be looking at it.