

CIS 4004: Web Based Information Technology Spring 2011

Advanced XHTML & CSS – Tables

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
<http://www.cs.ucf.edu/courses/cis4004/spr2011>

Department of Electrical Engineering and Computer Science
University of Central Florida



Designing Interface Components - Tables

- In HCI and interface component is what the user utilizes to interact with your webpage. In a sense, an interface component encompasses all the supporting elements for your page's content: links, menus, forms, tables, and so on.
- We'll be examining how these components should be written in XHTML to be valid and accessible, and how to style them with valid CSS in a variety of ways to meet the design needs of your projects.
- We'll start with tables.



Tables In XHTML and CSS

- As we've mentioned before, in the past, tables were a widely abused technique for creating page layouts full of presentation markup. CSS has provided a much cleaner and more adaptable method for accomplishing page layouts, so do not use tables as a technique for designing page layouts.
- However, tables should still be used for their designated purpose – laying out grids of data in rows and columns in the same manner in which it appears in spreadsheets or other applications where tabular data is required.



Tables In XHTML and CSS

- Tables in XHTML work much the same way they do in a spreadsheet or word processor application and resemble a matrix (grid).
- The entire table in XHTML is surrounded by the start table `<table>` tag and the end table tag `</table>`. You choose how many rows and columns you need for your table.
- The main body of a table is made up of rows and columns, like a matrix.
- In XHTML, you construct your tables one row at a time. Each row begins with an opening **table row** `<tr>` tag. Each column within that row contains the open and end tags for the **table data** `<td>` element.



Tables In XHTML and CSS

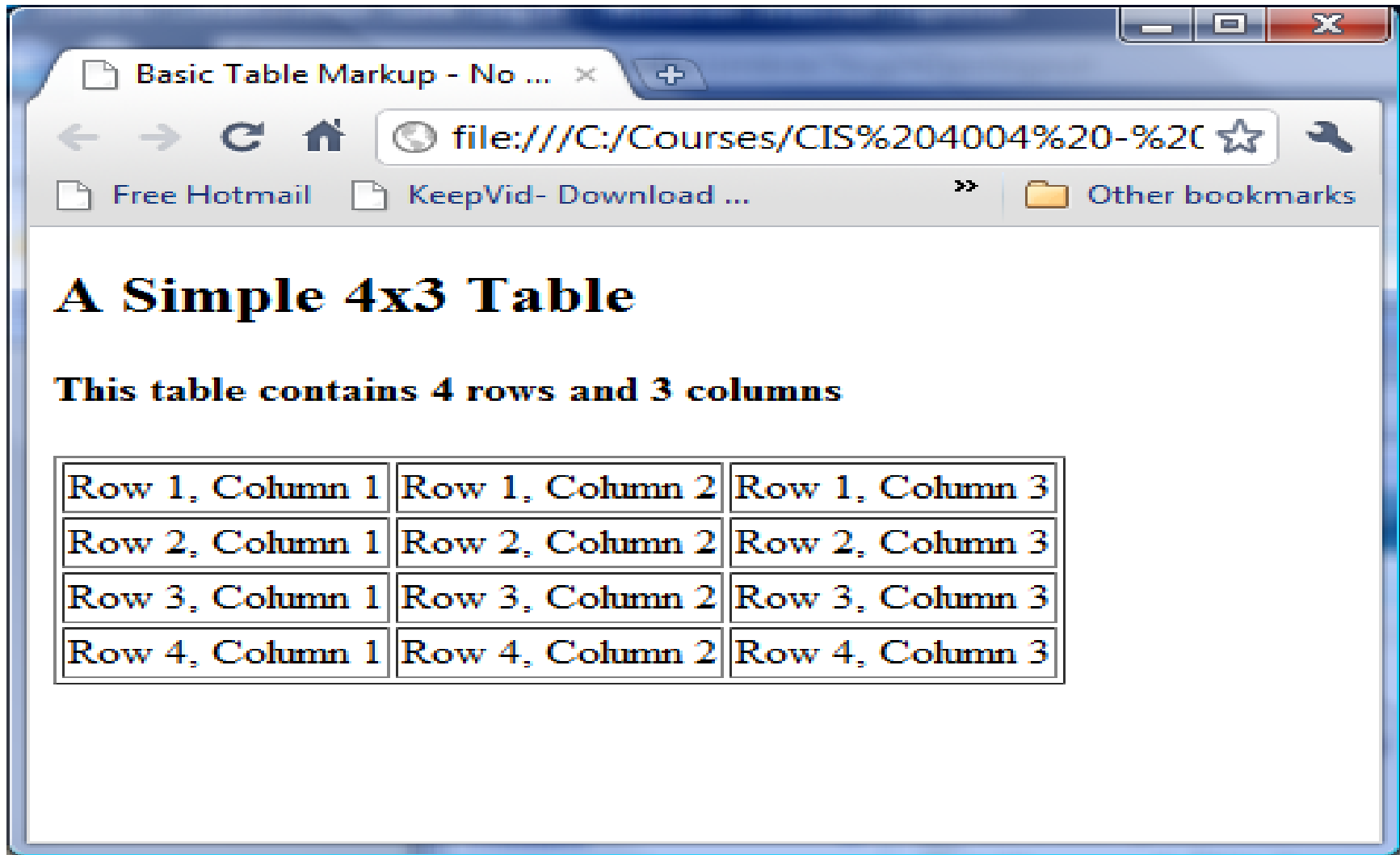
- After the last `<td>` element is closed for the last column, the row ends with the ending `</tr>` tag.
- This is the definition for a row that contains three columns.

```
<tr>  
  <td> column 1 </td>  
  <td> column 2 </td>  
  <td> column 3 </td>  
</tr>
```

- Shown on the next page is the markup for a simple XHTML table.



Tables In XHTML and CSS



The screenshot shows a web browser window with a single tab titled "Basic Table Markup - No ...". The address bar contains the file path "file:///C:/Courses/CIS%204004%20-%2C". Below the address bar are several bookmarks: "Free Hotmail", "KeepVid- Download ...", and "Other bookmarks". The main content area of the browser displays the following text and table:

A Simple 4x3 Table

This table contains 4 rows and 3 columns

Row 1, Column 1	Row 1, Column 2	Row 1, Column 3
Row 2, Column 1	Row 2, Column 2	Row 2, Column 3
Row 3, Column 1	Row 3, Column 2	Row 3, Column 3
Row 4, Column 1	Row 4, Column 2	Row 4, Column 3



```
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
notes.html notes.html config.txt first table example.html
6 <title>Basic Table Markup - No CSS</title>
7 </head>
8 <body>
9 <h2>A Simple 4x3 Table</h2>
10 <h4>This table contains 4 rows and 3 columns</h4>
11 <table border="1">
12 <tr>
13 <td>Row 1, Column 1</td>
14 <td>Row 1, Column 2</td>
15 <td>Row 1, Column 3</td>
16 </tr>
17 <tr>
18 <td>Row 2, Column 1</td>
19 <td>Row 2, Column 2</td>
20 <td>Row 2, Column 3</td>
21 </tr>
22 <tr>
23 <td>Row 3, Column 1</td>
24 <td>Row 3, Column 2</td>
25 <td>Row 3, Column 3</td>
26 </tr>
27 <tr>
28 <td>Row 4, Column 1</td>
29 <td>Row 4, Column 2</td>
30 <td>Row 4, Column 3</td>
31 </tr>
32 </table>
33 </body>
34 </html>
35
```

The border attribute is used to define the border of all elements inside the table. The "1" means a 1 pixel wide default style border is applied. Without this, there will be no gridlines at all in or around the table. We'll see additional styling with CSS later, but this is a minimal style for a table.



Formatting Tables

- Labeling table sections with `<thead>`, `<tfoot>`, and `<tbody>`.
- The `<thead>`, `<tfoot>`, and `<tbody>` can be used to define logical sections of a table. These elements are used to group the various rows in a table into a header (`<thead>`), body (`<tbody>`), and footer (`<tfoot>`) section. While not heavily used in practice today these elements will become more important as new user agents become more dependent on document structure, so you want to be sure to use them when necessary. For example, talking screen readers.
- These elements are optional, but when used must appear in the following order: `<thead>`, `<tfoot>`, `<tbody>`. The `<tfoot>` element must appear after the ending tag for the `</thead>` and before the open tag for the `<tbody>` element, even though its content will be displayed at the bottom of the table in a browser.



Formatting Tables

- The various table elements have a number of attributes that can be used to customize the look and layout of tables, rows, and cells.
- As with other elements, XHTML Strict does not allow all of the formatting attributes that Transitional and Frameset allow.
- The next few pages illustrate the most common attributes that can be used with the `<table>`, `<th>`, `<tr>` and `<td>` elements.



<table> Element Attributes

<u>Name</u>	<u>Description and Values</u>
summary	Text description of the table. Useful for non-visual browsers.
width	Sets the width of the table. Values: Percentage or pixels
border	Sets the width of the border around the table. Values: A value of 0 makes the border invisible. An integer value greater than 0 will result in a border of that number of pixels.
cellpadding	Sets the amount of space between the border of the table cell and the data contained in the cell. Values: Percentage or pixels
cellspacing	Sets the amount of space between cells. Values: Percentage or pixels
frame	Defines which sides of the table will be displayed. Values: above, below, border, box, lhs, bsides, rhs, vsides, void
rules	Defines which rule lines will be displayed. Values: all, cols, groups, none, rows



<td> And <th> Element Attributes

<u>Name</u>	<u>Description and Values</u>
align	Horizontal alignment of data in a cell Values: left, center, right, justified
valign	Vertical alignment of data in a cell Values: top, middle, bottom
rowspan	Number of rows a cell spans Values: integer greater than 1 and less than or equal to the total number of rows in the table
colspan	Number of columns a cell spans Values: integer greater than 1 and less than or equal to the total number of columns in the table
abbr	Used for an abbreviated version of the content of the cell
axis	Used to assign a cell to a category group
headers	List of cells that provide header information for the current cell based on the values of the id attributes of the header cells. This list is space delimited.
scope	Provides information about which cells the current header cell provides header information for Values: col, colspan, row, rowspan



<tr> Element Attributes

<u>Name</u>	<u>Description and Values</u>
<code>align</code>	Horizontal alignment of data in all cells in a row Values: left, center, right, justified
<code>valign</code>	Vertical alignment of data in all cells in a row Values: top, middle, bottom



XHTML Table – Examples

- For our second table example, we'll have content that spans multiple rows and columns.
- As with any skill, the best way to master the skill is to practice, practice, practice, so I encourage you to try and develop some additional tables on your own. I've put a couple of practice problems at the end of this set of notes for you to try.



```

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
notes.html notes.html config.txt first table example.html second table example.html
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <title>Content Spanning Multiple Table Rows and Columns with XHTML Tables</title>
7 <style type="text/css">
8 <!--
9     #teal_bg { background-color: #99FFFF;
10    }
11    .yellow_bg { background-color: #FFFF00;
12    }
13    .blue_bg {background-color: blue;
14              color:white;
15    }
16    #red_bg {background-color:#F73;
17    }
18    caption { font-weight: bold;
19              font-size: 14pt;
20              text-align: center;
21              color: #000099;
22    }
23    -->
24 </style>
25 </head>
26 <body>
27 <!-- Begin Table -->
28 <table border="1" cellpadding="5px" id="teal_bg">
29 <caption>Saltwater Aquarium Invoice</caption>

```



File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

notes.html notes.html config.txt first table example.html second table example.html

```

26 <body>
27   <!-- Begin Table -->
28   <table border="1" cellpadding="5px" id="teal_bg">
29     <caption>Saltwater Aquarium Invoice</caption>
30     <thead>
31       <!-- Begin Header Row -->
32       <tr class="blue_bg">
33         <th rowspan="2">Item</th>
34         <th colspan="2">Purchase Details</th>
35         <th rowspan="2">Total Price</th>
36       </tr>
37       <tr class="blue_bg">
38         <th>Price</th>
39         <th>Quantity</th>
40       </tr>
41     </thead>
42     <tfoot>
43       <tr align="center" id="red_bg">
44         <td colspan="4"><small>Thank you for shopping with us.</small></td>
45       </tr>
46     <!-- End First Row -->
47     </tfoot>
48     <tbody>
49     <!-- Begin First Item -->
50     <tr>
51       <th>Blue Angel Fish</th>
52       <td align="center">$19.95</td>
53       <td align="center">2</td>
54       <td align="center">$39.90</td>

```

nb char : 2215 nb line : 82 Ln:1 Col:1 Sel:0 Dos\Windows ANSI INS



```

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
notes.html notes.html config.txt first table example.html second table example.html
52         <td align="center">$19.95</td>
53         <td align="center">2</td>
54         <td align="center">$39.90</td>
55     </tr>
56 <!-- End First Item -->
57 <!-- Begin Second Item -->
58     <tr>
59         <th>Sailfin Tang Fish</th>
60         <td align="center">$34.95</td>
61         <td align="center">1</td>
62         <td align="center">$34.95</td>
63     </tr>
64 <!-- End Second Item -->
65 <!-- Begin Third Item -->
66     <tr>
67         <th>Clown Fish</th>
68         <td align="center">$3.95</td>
69         <td align="center">4</td>
70         <td align="center">$15.80</td>
71     </tr>
72 <!-- End Third Item -->
73 <!-- Begin Invoice Total -->
74     <tr class="yellow_bg">
75         <td colspan="3" align="right"><b>TOTAL</b></td>
76         <td align="center"><b>$89.75</b></td>
77     </tr>
78 <!-- End Invoice Total -->
79 </tbody>
80 </table>

```

nb char : 2215 nb line : 82

Ln : 1 Col : 1 Sel : 0

Dos\Windows ANSI

INS



Content Spanning Multip... x

file:///C:/Courses/CIS%204004%20-%20%20

Free Hotmail KeepVid- Download ... Other bookmarks

Saltwater Aquarium Invoice

Item	Purchase Details		Total Price
	Price	Quantity	
Blue Angel Fish	\$19.95	2	\$39.90
Sailfin Tang Fish	\$34.95	1	\$34.95
Clown Fish	\$3.95	4	\$15.80
TOTAL			\$89.75

Thank you for shopping with us.



XHTML Table – Examples

- The third table example illustrates a nested table (a table within a table).



```
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
notes.html notes.html config.txt first table example.html second table example.html table example 3.html
17 <!-- Begin Outer Table -->
18 <table border="1" cellpadding="5" class="outer">
19 <caption><strong>Phone Book</strong></caption>
20 <!-- Begin Header Row -->
21 <tr>
22 <th>Name</th>
23 <th>Address</th>
24 <th>Phone Number</th>
25 </tr>
26 <!-- End Header Row -->
27 <!-- Begin First Row -->
28 <tr>
29 <td>Eva Mendes</td>
30 <td>
31 <!-- Begin Inner Table -->
32 <table border="1" class="inner">
33 <tr>
34 <td colspan="3">123 West Liberty Dr.</td>
35 </tr>
36 <tr>
37 <td>Miami</td>
38 <td>FL</td>
39 <td>32769</td>
40 </tr>
41 </table>
42 <!-- End Inner Table -->
43 </td>
44 <td>(321) 555-1212</td>
45 </tr>
```

Inner table starts here

Inner table ends here



Content Spanning ... x Spanning Multiple ... x

file:///C:/Courses/CIS%204004%20-%2C

Free Hotmail KeepVid- Download ... Other bookmarks

Phone Book

Name	Address	Phone Number
Eva Mendes	123 West Liberty Dr. Miami FL 32769	(321) 555-1212
Carrie Underwood	456 Lighthouse Way Nashville TN 02901	(415) 555-1212



Designing Interface Components - Forms

- Forms are important to many web sites because they provide a means by which user-entered data can be sent from the browser across the network to the web site's web server (e.g. eustis.eecs.ucf.edu).
- From simple log-ins and sign-ups, to multi-page e-commerce checkouts, forms are everywhere, and understanding how to create them is a key skill that every web developer should have.
- Since forms are such an important part of many websites, we'll look at both how forms work and how they are marked up in XHTML as well as styling them with CSS.



Designing Interface Components - Forms

- Every time you use a search engine, place an order, or join an online mailing list, you use a **form**.
- A form is an XHTML element that contains and organizes other objects – such as text boxes, check boxes, and buttons – that can accept information from Web site visitors.

This form has only three elements. One text box that accepts the keywords to be used in the search, and two buttons that can submit the form and get the search started.



How Forms Work

- The purpose of a form is to gather various pieces of data from the user.
- When the form is submitted, usually by clicking a button on the screen, the form data, structured as a set of name/value pairs is passed to the server to be processed by a server-side script written in a language such as PHP, Java, or Perl.
- Even though you might not be the person who is writing the server-side code, you need to know how to create the forms in XHTML so that they send the correctly structured data to the server for processing.



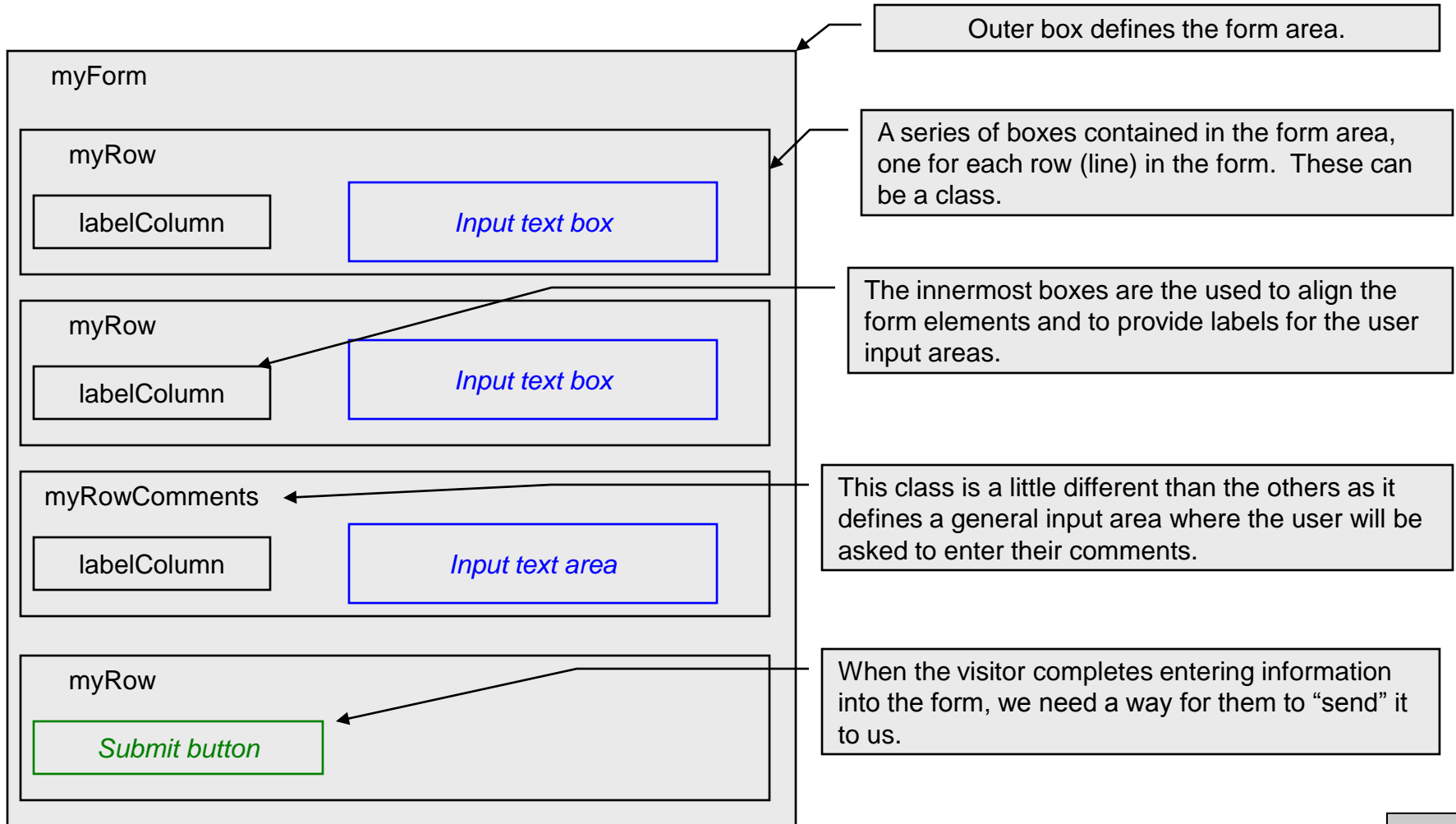
Using CSS to Style a Form

- Tables have been traditionally used to create forms in Web pages.
- Since you are learning the new style of Web page development in this course which utilizes CSS for all page layout, we'll also learn the new way of creating forms using only CSS without tables.
- To create a form using only CSS, you take advantage of the CSS box model to create a series of boxes which are the elements (rows and columns) of your table.
- As we've done with our splash pages and content pages, the best way to design a form using CSS is to first storyboard the form layout and then create the XHTML document to include the form elements and finally create the CSS to render the form with the correct layout.



Using CSS To Style The Form

- Shown below is the storyboard for the form we want to create.



Form without CSS

First Form Example - No ...

file:///C:/Courses/CIS%204004%20-%20Web%20Based%20IT/Spring%202011/cc

Free Hotmail KeepVid- Download ... Suggested Sites Web Slice Gallery Other bookmarks

Input text box

Tell Us What You Think About CIS 4004!

Name:

Input text area

E-mail:

The submit button

Comments:



Markup Only

```

1  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml" >
5  <head>
6     <title>First Form Example - No CSS</title>
7  </head>
8  <body >
9  <h2>Tell Us What You Think About CIS 4004!</h2>
10 <div>
11 <form method="post" action="mailto:markl@cs.ucf.edu" enctype="text/plain" >
12 <div>
13 <label for="Name"></label>
14 <span class="labelCol">Name:</span>
15 <span><input type="text" name="Name" id="Name" /></span>
16 </div>
17 <div>
18 <label for="Email"></label>
19 <span class="labelCol">E-mail:</span>
20 <span><input type="text" name="Email" id="Email" /></span>
21 </div>
22 <div>
23 <label for="Comments"></label>
24 <span class="labelCol">Comments:</span>
25 <textarea name="Comments" id="Comments" rows="10" cols="75"></textarea>
26 </div>
27 <div>
28 <input id="submitButton" type="submit" value="Send Your Comments" />
29 </div>
30 </form>

```



New XHTML Tags: <form>

- A form in XHTML is contained within a `form` element. The form itself contains regular text and other XHTML form elements such as check boxes, pull-down menus, and text fields.
- The W3C specification calls these form elements **controls**, but are most commonly referred to as “**form fields**”.
- There are a variety of form fields that can be inserted.
- In order to make a form work, you will need to specify two things in the form tag: (1) the address of the program that will handle the form contents using the `action` attribute and (2) the method by which the form data will be passed using the `method` attribute.
- The `action` attribute determines how the form is to be handled. The action attribute is usually set to the URL of the program that will handle the data in the form. We’ll discuss this attribute in more detail later but for now we won’t really “handle” the data in the form in any significant manner. In our example, we will simply email the data.



New XHTML Tags: `<form>`

- The `method` attribute determines how the form will be submitted to the address specified by the `action` attribute.
- There are only two acceptable values for this attribute: `get` and `post`. These are HTTP methods that a browser uses to talk to a server. (See Introduction – Part 2, page 33.) We won't worry about the subtle differences between these two values for the moment, but in general `post` will be preferred whenever either large amounts of data are being sent to the server or data is to be stored in a database.
- The `enctype` attribute determines how the data in the form is to be encoded when it is sent to the server. The default for most browsers is the same MIME format that is used for URLs. Particularly when using a `mailto` URL, the encoding type `text/plain` is normally used for ease of reading.



New XHTML Tags: `<form>`

- A web page can contain any number of form elements, but forms cannot be nested inside one another, unlike list elements which can.
- This is because each form's contents must be handled separately by the action attribute of that form, so nesting of forms is not possible.
- We'll see later how to group elements of a form together to enhance the visual clarity of the form, and while this may give the illusion of a nested form, rest assured that it is not nested.
- What we'll do for the rest of this section of notes is two things: (1) we'll add CSS styling to our forms and (2) look at the various form controls in XHTML to add features to the form.



Using CSS To Style The Form

- We'll define an `id` called `myForm` to define the properties of the entire form area, thus, `myForm` is the outermost box.
- The `myRow` class will set the height for a typical line in the form.
- The `myRowComments` class will configure a form area that is a general text input area where we will receive comments from the visitors. We'll ultimately set this up so that a 20 pixel margin is set below the scrolling text box.
- The `labelCol` class is the key to aligning the text. We'll need to set the width of this element to accommodate our largest label. We'll start out with 100 pixel width and adjust from there. All of the text in the `labelCol` elements will align to the right.
- We'll create the markup first and then worry about applying the styles.



Tell Us What You Think About CIS 4004!

Name:

E-mail:

Comments:

CSS for the form

```
7 <style type="text/css">
8 <!--
9     body{ font-family:Arial,sans-serif;
10         background-color:#EEE; /*was DDD*/
11         color:#000000;
12     }
13     #myForm {border:3px solid black; /*was blue */
14         padding:10px;
15         margin:10px;
16         background-color: #CCC;
17         width: 750px;
18     }
19     .myRow {height:30px;
20     }
21     #myRowComments {margin-bottom:20px;
22     }
23     .labelCol{float:left; width: 100px; text-align: right; padding: 0px 8px 0px 0px;
24     }
25     #submitButton { background-color:green; color:white;
26     }
27     -->
28 </style>
29 </head>
30 <body >
```



Browser Rendering Of The Form

First Form Example With ... x

file:///C:/Courses/CIS%204004%20-%20Web%20Based%20IT/Spring%202011/co

Free Hotmail KeepVid- Download ... Suggested Sites Web Slice Gallery

Tell Us What You Think About CIS 4004!

Name:

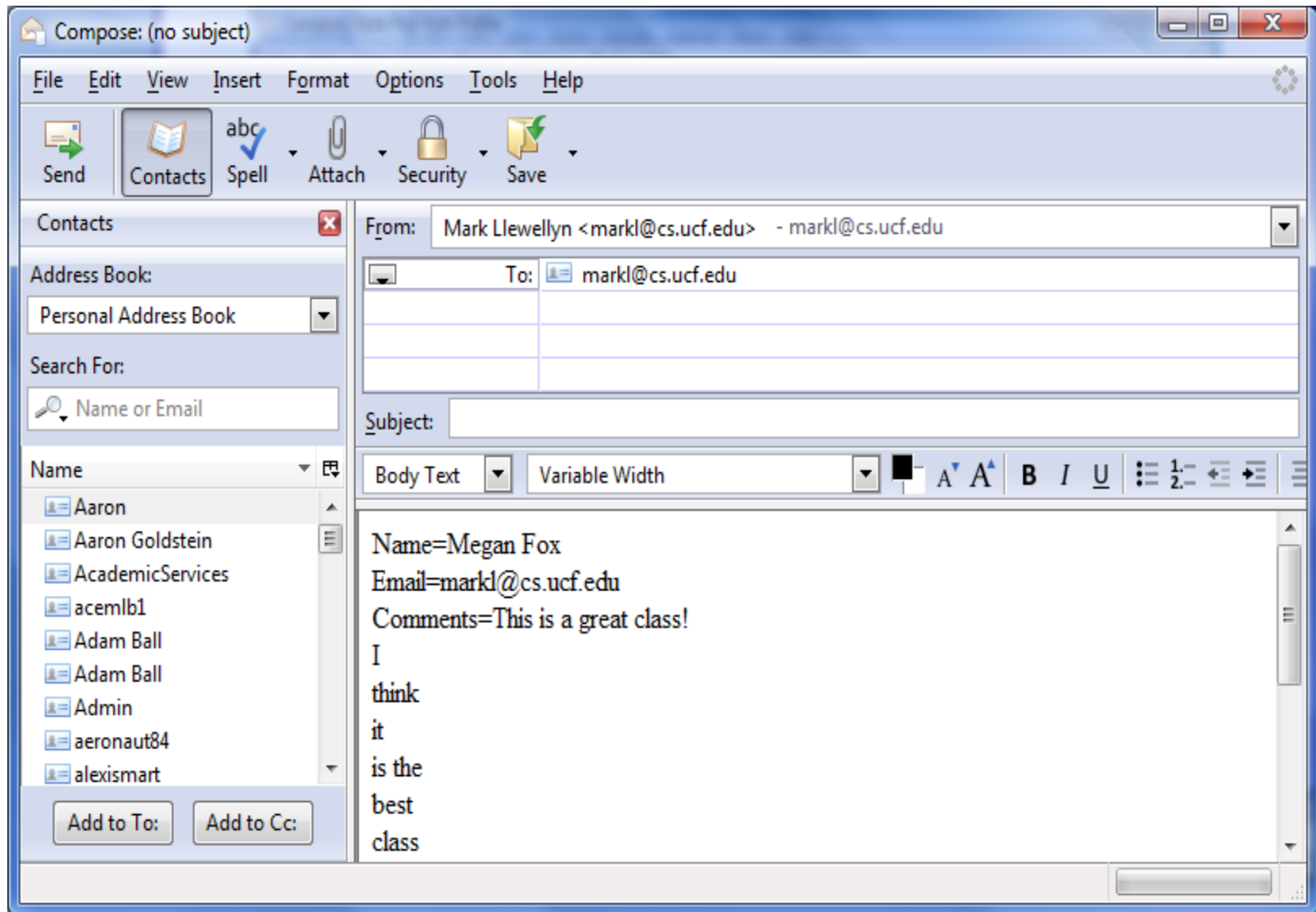
E-mail:

Comments:

Notice the up/down sliders appear when the entered text expands beyond the visible portion.★



Email program activated by clicking button



https://rockwell.cs.ucf.edu/zimbra/#? Email received by markl@cs.ucf.edu

File Edit View Favorites Tools Help

Google Search Share Check AutoFill Sign In

CIS 4004 Lecture Notes Zimbra: Inbox (1399)

Zimbra Email Search Save Advanced Learn about the offline version Help Log Out

Dr. Mark Llewellyn 1.8 GB of unlimited

Folders

- Inbox (1399)
- Sent
- Drafts
- Junk
- Trash
- Advising
- CGS 2545 - Spring 20
- CGS 2585 - Spring 20
- CGS 3175
- CNT 4714

January 2011

S	M	T	W	T	F	S
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Mail Address Book Calendar Tasks Documents Briefcase Preferences

New Get Mail Delete Reply Junk View 100+ messages

	From	Subject	Folder	Size	Received
<input type="checkbox"/>	Dr. Ali Orooji	Re: Doing ok? - Hi Andrew / Mark, Dr. Bauer needs to hire ...	Inbox	5 KB	12:35 PM
<input type="checkbox"/>	Mark Llewellyn	From Megan Fox! - Name=Megan Fox Email=markl@cs.ucf.edu	Inbox	2 KB	12:35 PM
<input type="checkbox"/>	Nano Materials (...)	Seminar-01/21/2011-Dr. Liangfeng Sun, Creol roo...	Inbox	6 KB	12:30 PM

From Megan Fox! January 20, 2011 12:42 PM

From: "Mark Llewellyn" <markl@eecs.ucf.edu>
To: mark

Name=Megan Fox
Email=markl@cs.ucf.edu
Comments=This is a great class!
I think it is the best class

Done Internet | Protected Mode: Off 100%



New XHTML Tags:

- There are a number of basic form elements that are used to gather information from the Web site visitor. Some of these are:
 - Text boxes:
 - Scrolling text boxes
 - Select lists
 - Radio buttons
 - Check boxes
 - Buttons
- XHTML tags that configure these form elements include the `<input />`, `<textarea>`, `<select>`, and `<option>` tags.
- We'll look at a number of these form elements in detail now.

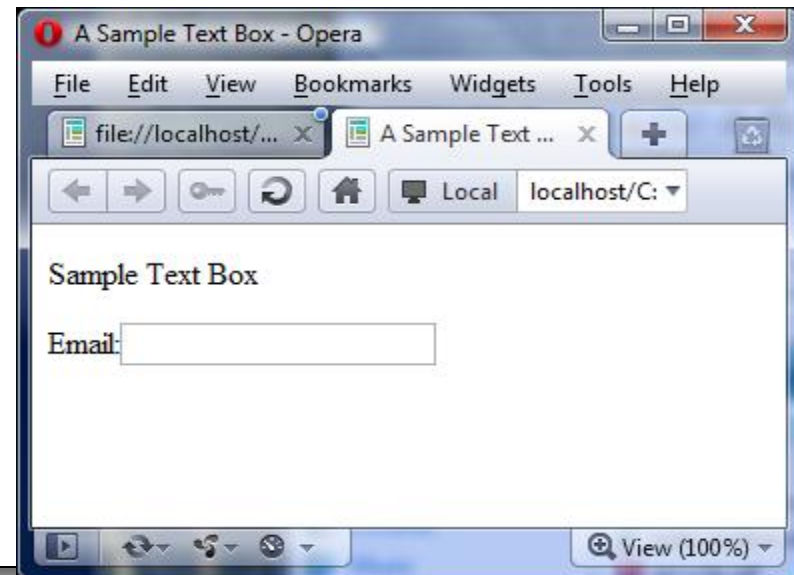
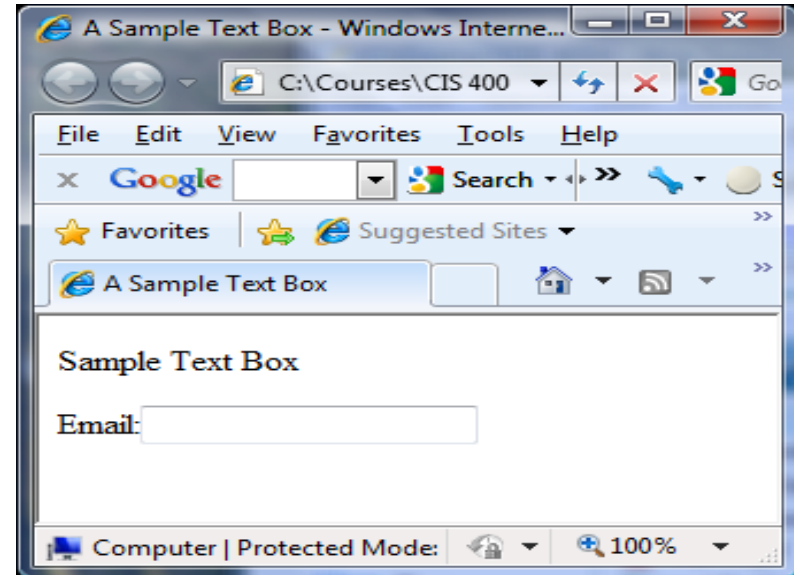


Form Element: Text Box

- Text boxes are configured by the `<input />` tag.
- Text boxes accept text or numeric information.

Example:

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Sample Text Box</title>
</head>
<body>
<p> Sample Text Box </p>
<form method="get"
action="mailto:mark1@cs.ucf.edu">
<p>Email:<input type="text" id="email" /></p>
</form>
</body>
</html>
```

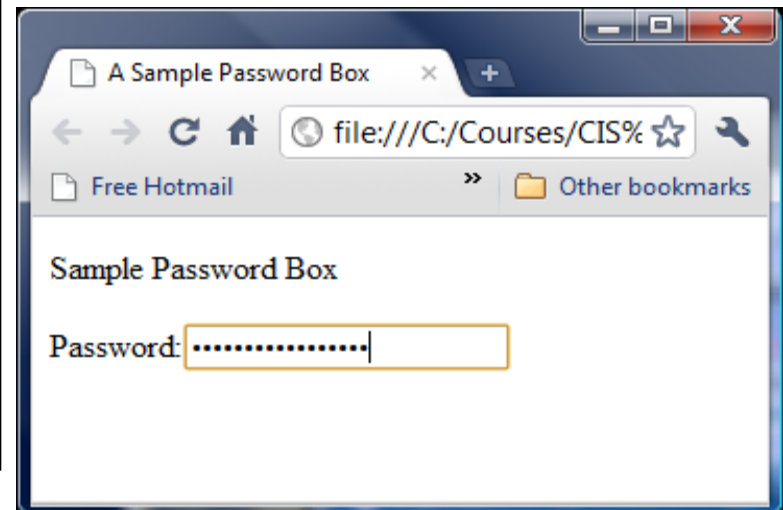
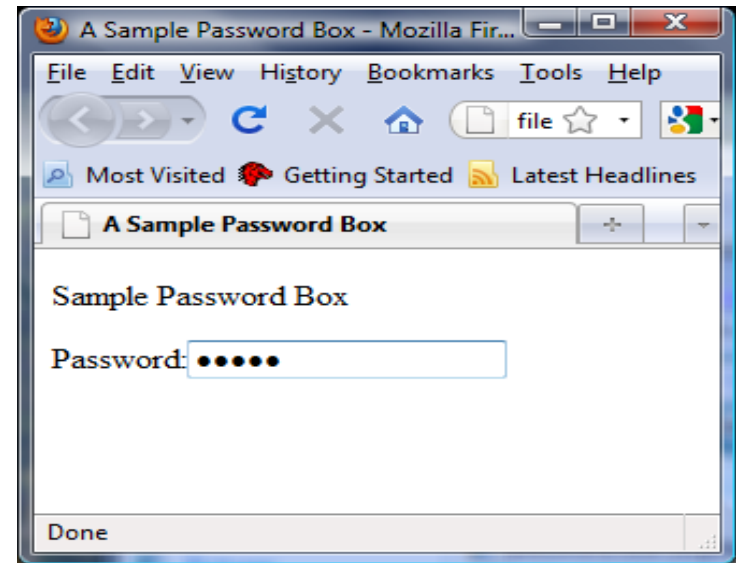


Form Element: Password Box

- Password boxes are also configured by the `<input />` tag.
- Password boxes are like text boxes except that they hide data as it is entered.

Example:

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Sample Password Box</title>
</head>
<body>
<p> Sample Password Box </p>
<form method="get"
action="mailto:mark1@cs.ucf.edu">
<p>Password:<input type="password" id="psswd"
/></p>
</form>
</body>
</html>
```



Attributes For Text and Password Boxes

Common Attributes	Values	Usage
type	"text", "password"	Configures the text box input type.
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
size	Numeric	Configures the width of the text box as displayed by the browser. If size is omitted, the browser displays the text box with its own default size.
maxlength	Numeric	Configures the maximum length of data accepted by the text box.
value	Text or numeric characters	Assigns an initial value to the text box that is displayed by the browser. Accepts the information typed in the text box. This value can be accessed by client-side scripting languages and by server-side processing.

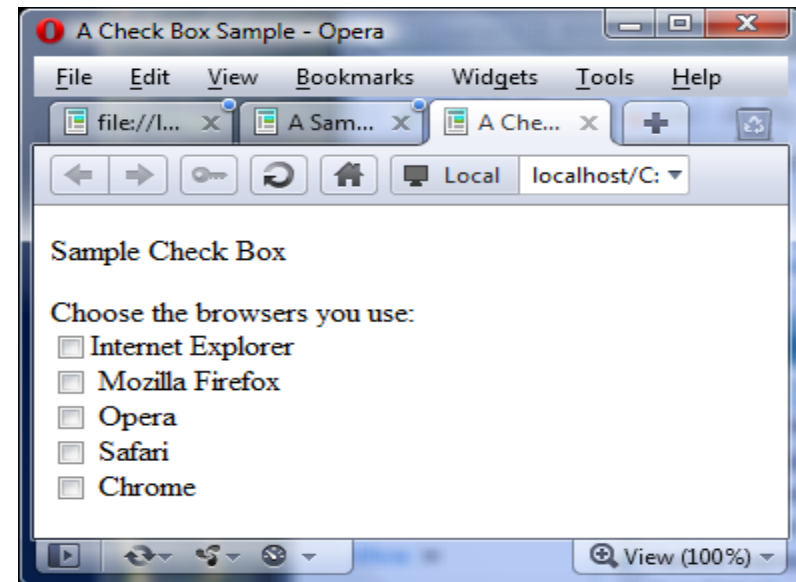
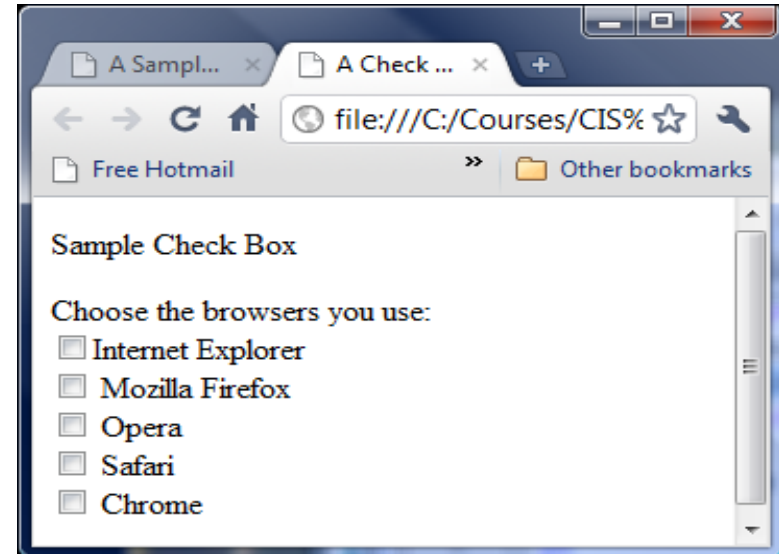


Form Element: Check Box

- Check boxes are also configured by the `<input />` tag.
- Check boxes allow the user to select one or more of a group of predetermined items.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Check Box Sample</title>
</head>
<body>
  <p> Sample Check Box </p>
  <form method="get" action="mailto:markl@cs.ucf.edu">
    <p>Choose the browsers you use:<br />
      <input type="checkbox" id="IE" value="yes" />Internet Explorer <br />
      <input type="checkbox" id="Mozilla" value="yes" /> Mozilla Firefox <br />
      <input type="checkbox" id="Opera" value="yes" /> Opera <br />
      <!-- more not shown here -->
    </p>
  </form>
</body>
</html>
```



Attributes For Check Boxes

Common Attributes	Values	Usage
type	"checkbox"	Configures the check box.
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
checked	"checkbox"	Configures the check box to be checked by default when displayed by the browser.
value	Text or numeric characters	Assigns a value to the check box that is triggered when the check box is checked. This value can be accessed by client-side and server-side processing.

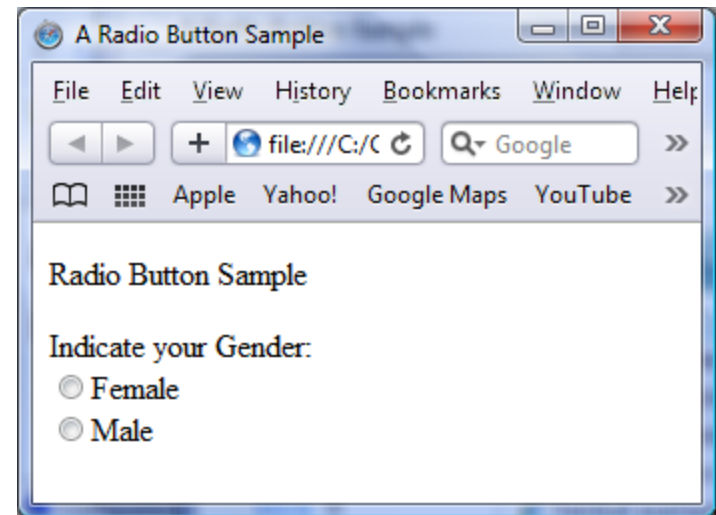
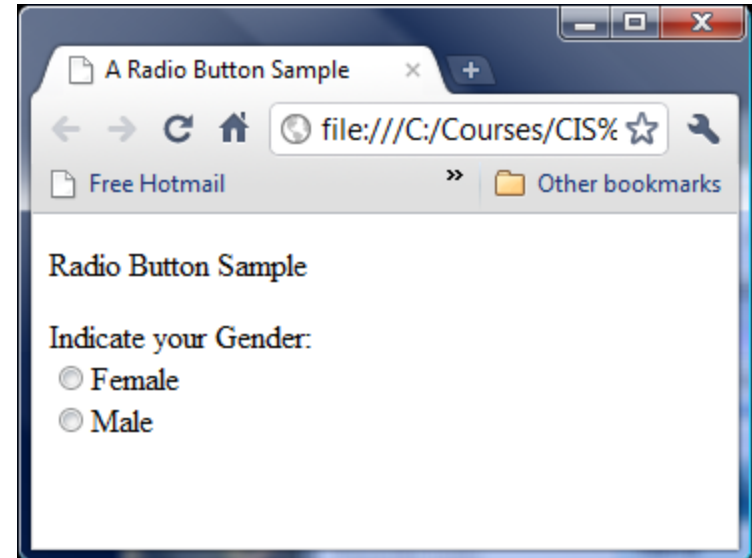


Form Element: Radio Button

- Radio buttons are also configured by the `<input />` tag.
- Radio buttons allow the user to select exactly one item from a group of predetermined items.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Radio Button Sample</title>
</head>
<body>
  <p> Radio Button Sample </p>
  <form method="get" action="mailto:markl@cs.ucf.edu">
    <p>Indicate your Gender:<br />
      <input type="radio" name="gender" id="gf"
value="female" />Female <br />
      <input type="radio" name="gender" id="gm"
value="male" />Male <br />
    </p>
  </form>
</body>
</html>
```



Attributes For Radio Buttons

Common Attributes	Values	Usage
type	"radio"	Configures the radio button.
name	Alphanumeric, no spaces, begins with a letter	REQUIRED – All radio buttons in a group must have the same name. This attributes also names the form element for access\by client and server side scripts.
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
checked	"checked"	Configures the radio button to be selected by default when displayed by the browser.
value	Text or numeric characters	Assigns a value to the radio button that is triggered when the radio button is checked. This should be a unique value for each radio button in a group. This value can be accessed by client-side and server-side processing.

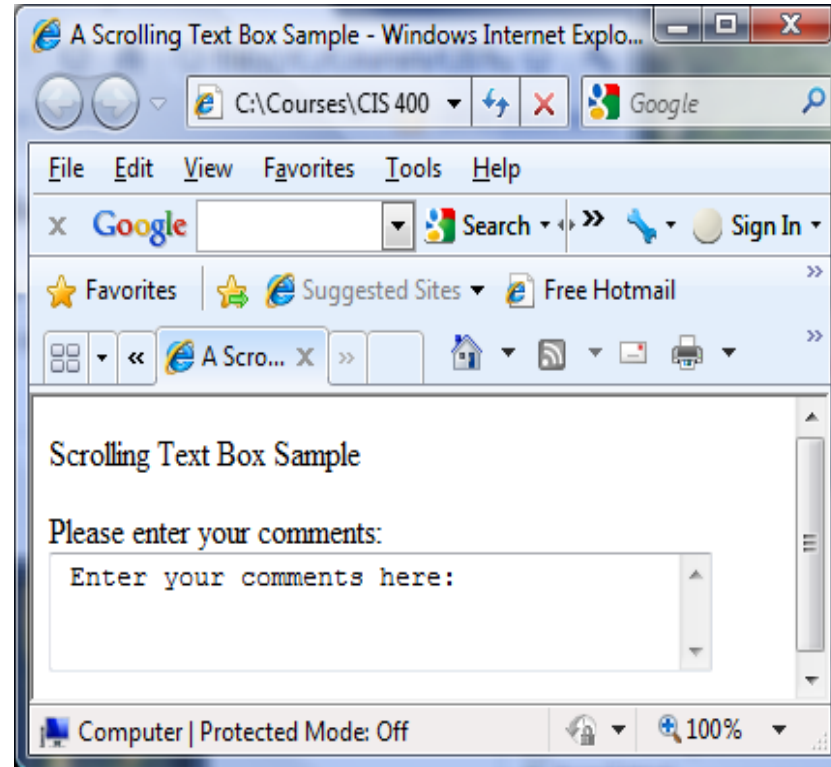


Form Element: Scrolling Text Box

- The `<textarea>` container tag configures a scrolling text box. A scrolling text box is used for accepting free-form comments, questions, or descriptions.

Example:

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Scrolling Text Box Sample</title>
</head>
<body>
<p> Scrolling Text Box Sample </p>
<form method="get"
action="mailto:mark1@cs.ucf.edu">
<p>Please enter your comments:<br />
<textarea id="comments" cols="40"
rows="3"> Enter your comments here:</textarea>
</p>
</form>
</body>
</html>
```



Attributes For Scrolling Text Boxes

Common Attributes	Values	Usage
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
cols	numeric	Configures the width in character columns of the scrolling text box. If <code>cols</code> is omitted, the browser displays the scrolling text box with its own default width.
rows	numeric	Configures the height in rows of the scrolling text box. If <code>rows</code> is omitted, the browser displays the scrolling text box with its own default height.

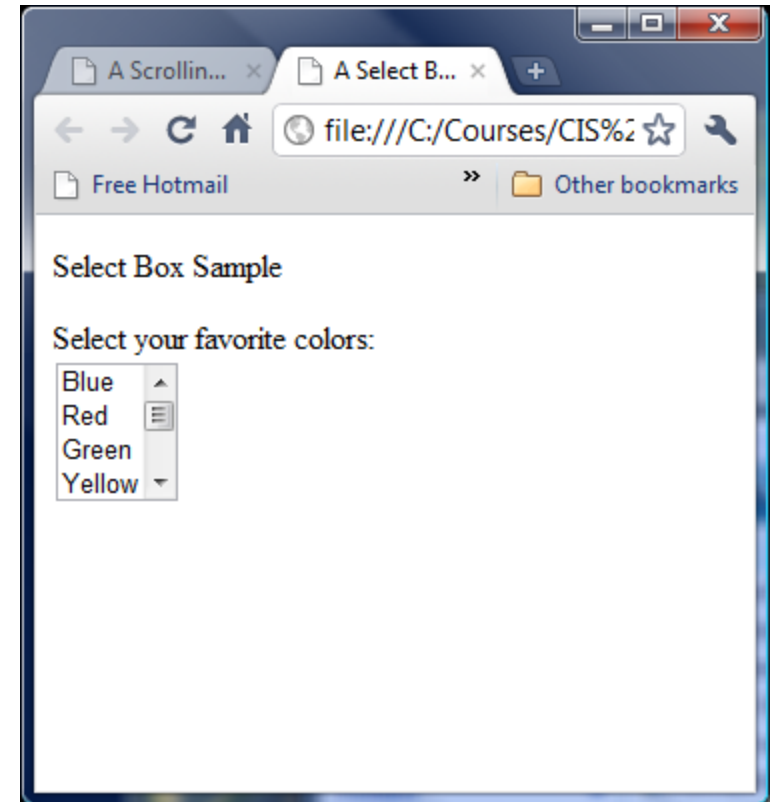


Form Element: Select List

- The `<select>` container tag (along with `<option>` tags) configures a select list. This form element has several names: select list, select box, drop-down list, drop-down box, and option box. It allows the visitor to select one or more items from a list of predetermined choices. The `<option>` container configures the choices in a select list.

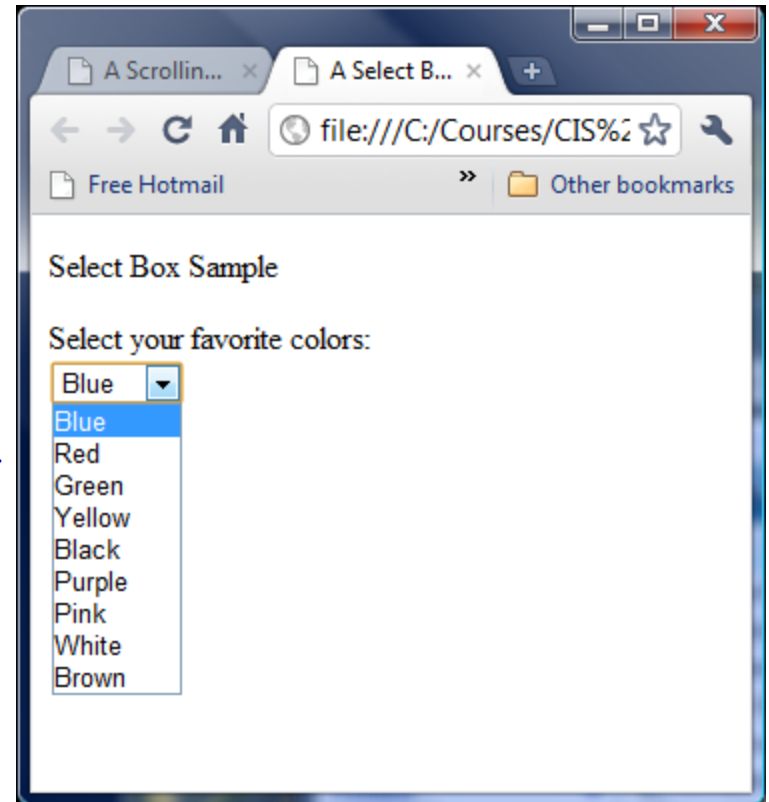
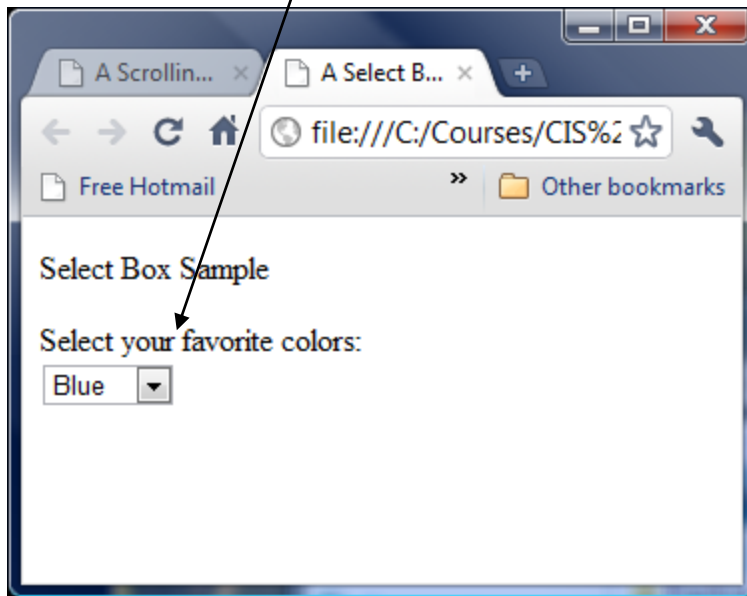
Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Select Box Sample</title>
</head>
<body>
  <p> Select Box Sample </p>
  <form method="get" action="mailto:markl@cs.ucf.edu">
    <p>Select your favorite colors:<br />
      <select id="favcolors" size="4">
        <option value="blue">Blue</option>
        <option value="red">Red</option>
        <option value="green">Green</option>
        <option value="yellow">Yellow</option>
        <option value="black">Black</option>
        <option value="purple">Purple</option>
        <option value="pink">Pink</option>
        <option value="white">White</option>
        <option value="brown">Brown</option>
      </select>
    </p>
  </form>
</body>
</html>
```



Form Element: Select List

With size set to 1 the select list will function as a drop-down box when the arrow is clicked.



Attributes For Select Lists

Common Attributes	Values	Usage
<select> tag		
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
size	numeric	Configures the number of choices the browser will display. If set to one, the element functions as a drop-down list. Scroll bars are automatically added by the browser if the number of options exceeds the space allowed.
rows	numeric	Configures the height in rows of the scrolling text box. If <code>rows</code> is omitted, the browser displays the scrolling text box with its own default height.
<option> tag		
value	Text or numeric characters	Assigns a value to the option. This value can be accessed by client-side and server-side processing.
selected	"selected"	Configures an option to be initially selected when displayed by a browser.

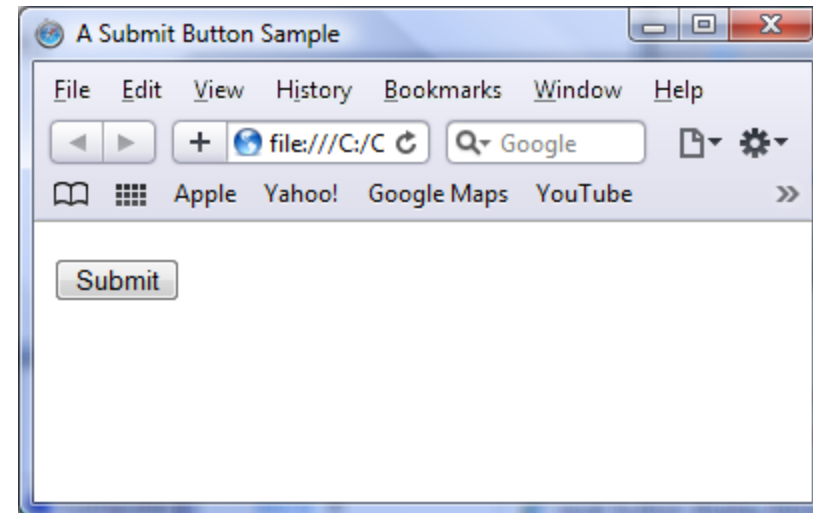
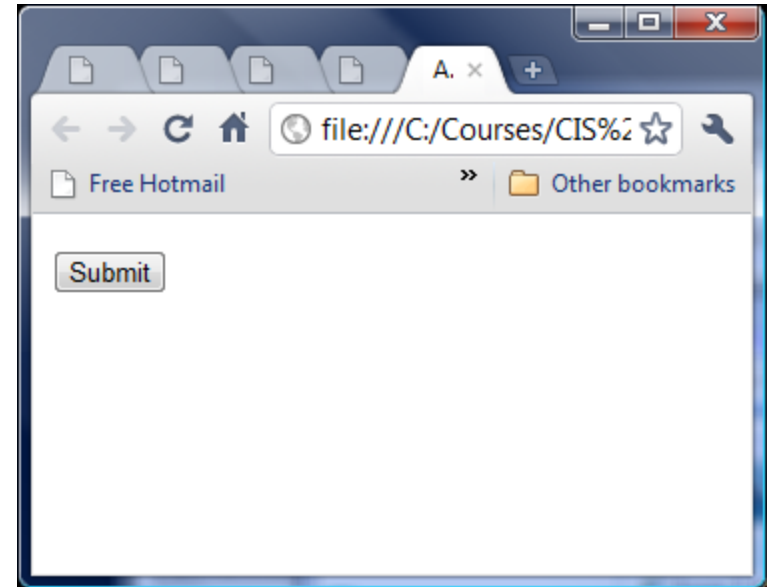


Form Element: Submit Button

- This form element is configured by the `<input />` tag and is used to submit the form. It triggers the action method on the `<form>` tag and causes the browser to send the form data to the Web server. The Web server will invoke the server-side processing listed on the form's action property.

Example:

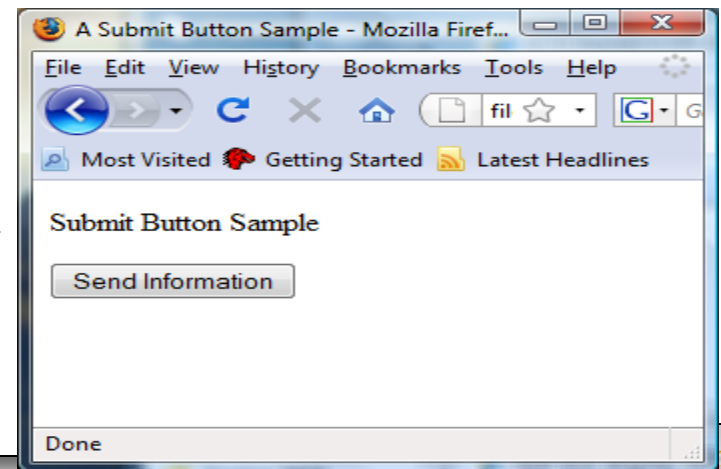
```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Submit Button Sample</title>
</head>
<body>
<p>Submit Button Sample </p>
<form method="get"
action="mailto:markl@cs.ucf.edu">
<p><input type="submit" /></p>
</form>
</body>
</html>
```



Attributes For Submit Buttons

Common Attributes	Values	Usage
type	"submit"	Configures the submit button.
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
value	Text or numeric characters	Configures the text displayed on the submit button. By default, the text "Submit Query" is displayed.

Example of a submit button with the value attribute set to "Send Information".

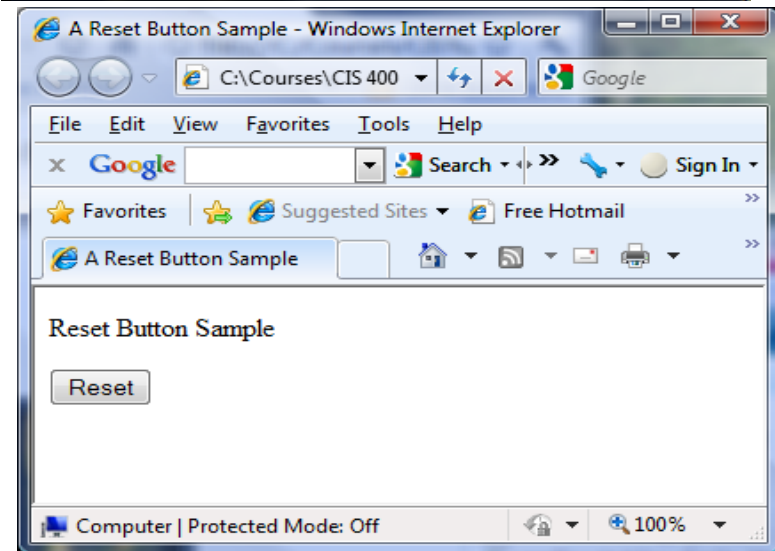
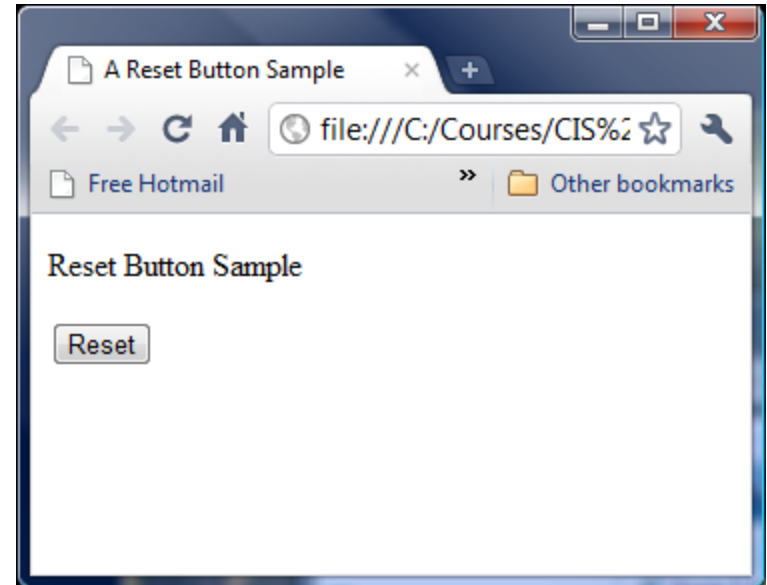


Form Element: Reset Button

- This form element is also configured by the `<input />` tag and is used to reset the form fields to their initial values.

Example:

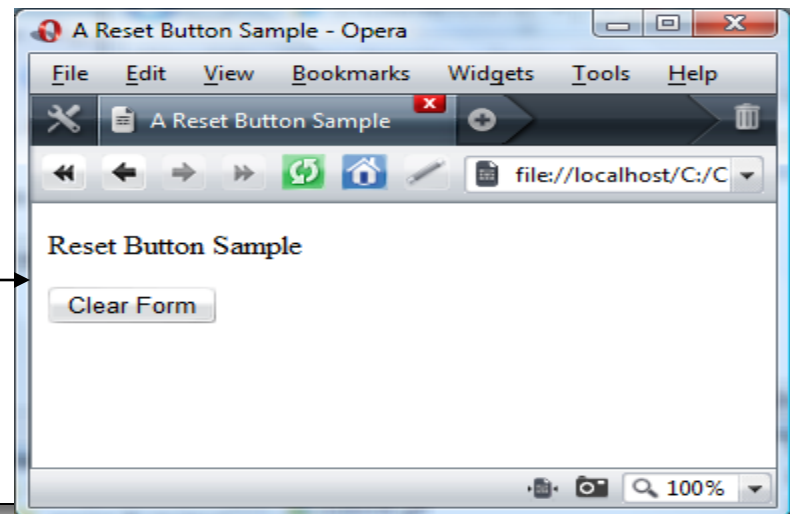
```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Reset Button Sample</title>
</head>
<body>
<p>Reset Button Sample </p>
<form method="get"
action="mailto:markl@cs.ucf.edu">
<p><input type="reset" /></p>
</form>
</body>
</html>
```



Attributes For Reset Buttons

Common Attributes	Values	Usage
type	"reset"	Configures the reset button.
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
value	Text or numeric characters	Configures the text displayed on the reset button. By default, the text "Reset" is displayed.

Example of a submit button with the value attribute set to "Clear Form".



Changing Input Properties

- You can change the default appearance of text boxes and password boxes by applying styling to the `<input>` tag.
- As with any property, the style property can be applied to tags in either external, internal, or inline (element by element) formats.
- Often you want different style properties to apply to similar elements in a single form, so it is common to apply styling to `<input>` tags on an element by element basis.
- Some examples are shown on the next several pages.

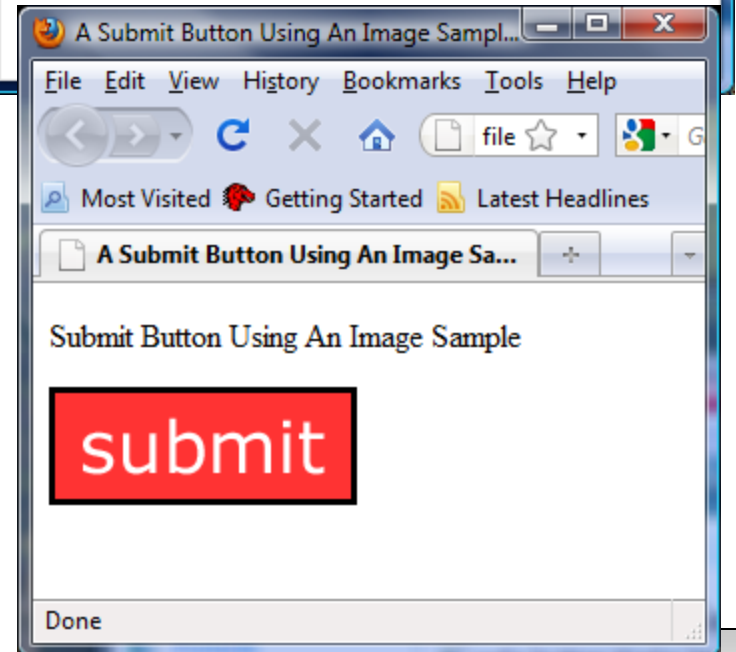
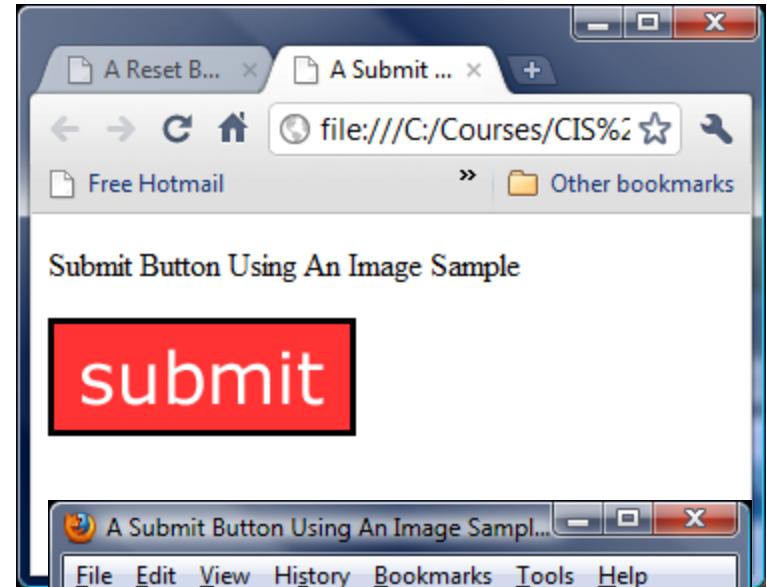


Using An Image For A Submit Button

- Using image as the type of input control allows you to use an image as a submit button.

Example:

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Submit Button Using An Image
Sample</title>
</head>
<body>
<p> Submit Button Using An Image Sample </p>
<form method="post"
action="mailto:markl@cs.ucf.edu">
<p><input type="image" src="submit.gif"
style="border:3px solid black" value="Send
Information" /></p>
</form>
</body>
</html>
```



Applying A Style To All `<input>` Tags

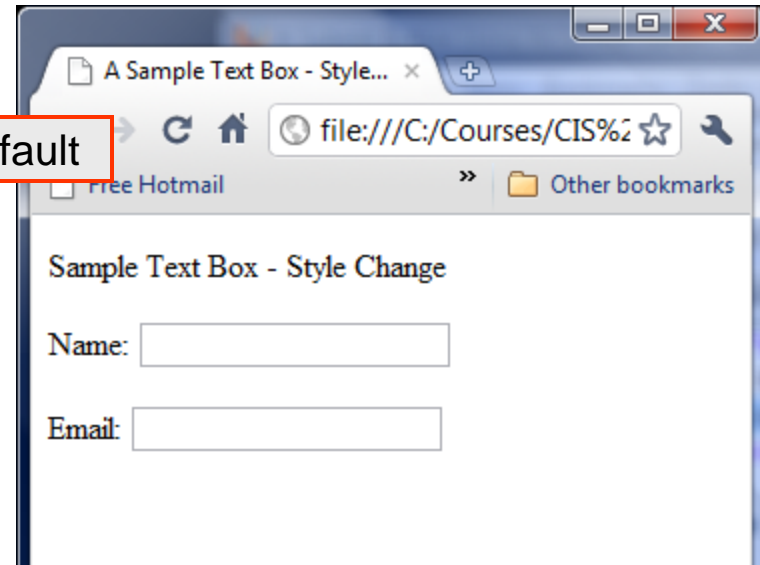
```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Sample Text Box - Style Change</title>

<style type="text/css">
  <!-- input { background-color:#00FFFF; }
  -->
</style>

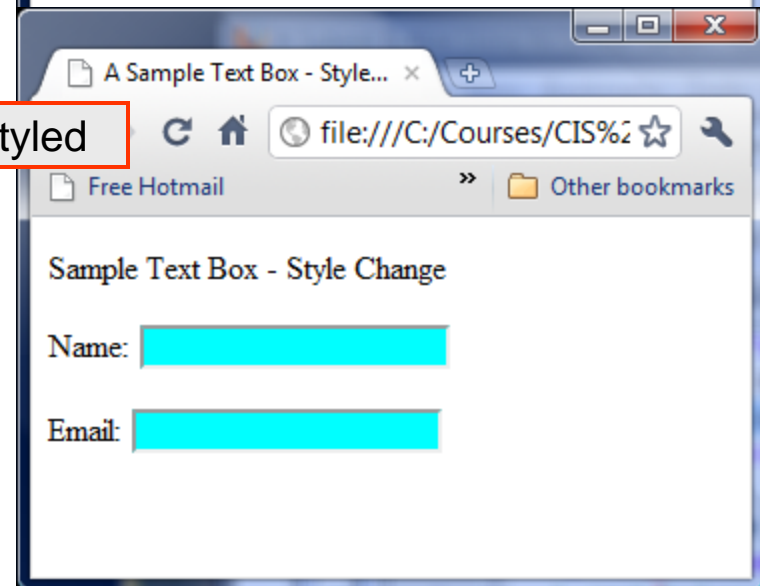
</head>
<body>
  <p> Sample Text Box - Style Change </p>
  <form method="get"
action="mailto:markl2@cs.ucf.edu">
    <p>Name: <input type="text" id="name" /></p>
    <p>Email: <input type="text" id="email" /></p>
  </form>
</body>
</html>
```

Change background-color for all `<input>` tags in this document.

default



styled



Applying A Style To All `<input>` Tags

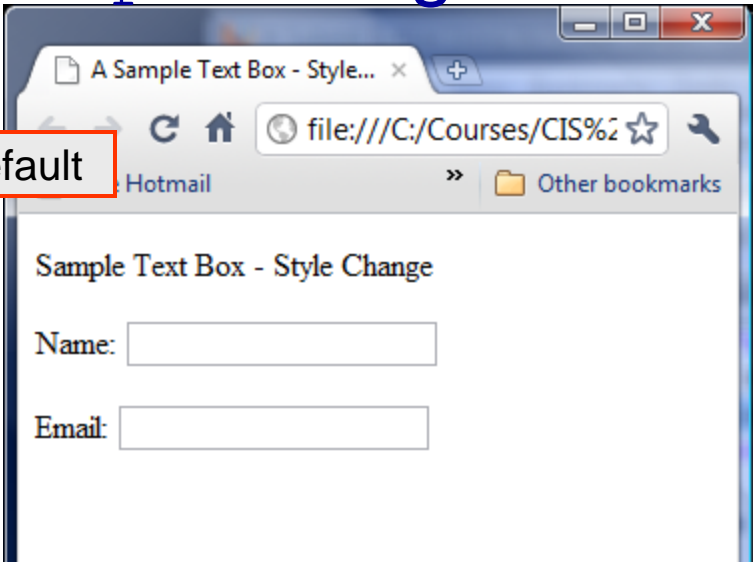
```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Sample Text Box - Style Change</title>

<style type="text/css">
  <!-- input { background-color:#00FFFF;
        border: solid 2px red; }
  -->
</style>

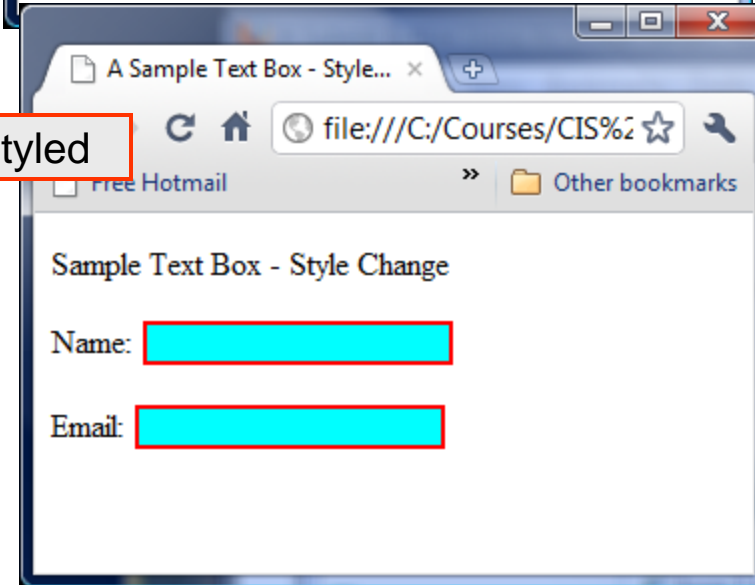
</head>
<body>
  <p> Sample Text Box - Style Change </p>
  <form method="get"
action="mailto:markl2@cs.ucf.edu">
    <p>Name: <input type="text" id="name" /></p>
    <p>Email: <input type="text" id="email" /></p>
  </form>
</body>
</html>
```

Set background-color to be light blue and set border to be 2 pixels in width, solid and red for all `<input>` tags in this document.

default



styled



Applying A Style To All <input> Tags

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Sample Text Box - Style Change</title>
</head>

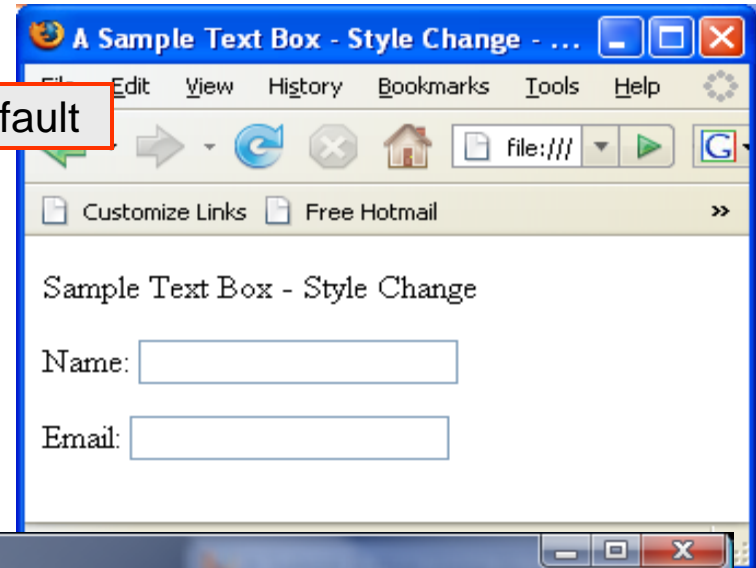
<body>
  <p> Sample Text Box - Style Change </p>
  <form method="get"
action="mailto:markl2@cs.ucf.edu">
    <p>Name: <input type="text" id="name"
style="background-color:#00FFFF; border:solid 2px
red" /></p>

    <p> Email: <input type="text" id="email"
style="background-color:#FFFF00; border:solid 2px
black" /></p>
  </form>
</body>
</html>
```

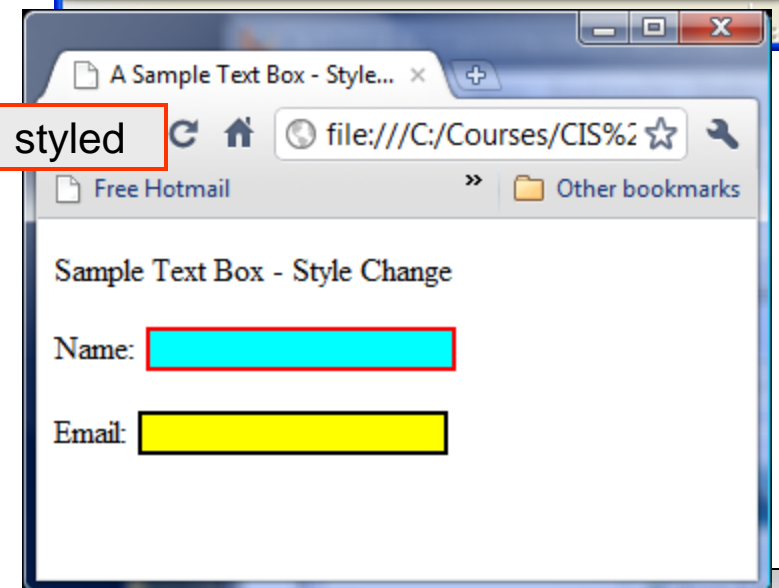
Style applied to this input tag sets a light blue background color and a red solid border.

Style applied to this input tag sets a yellow background color and a black solid border.

default



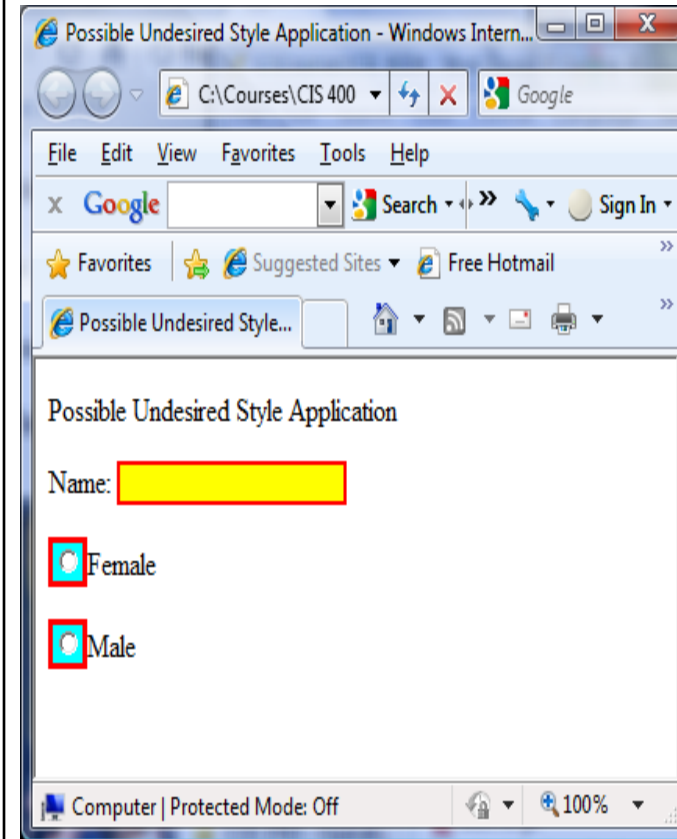
styled



CAUTION: Undesired Style Application

- When you apply global or document-wide styles to an input selector and the input selector is not defined in a class, all input elements on the page take on the properties defined in the input selector. For example, radio buttons and check boxes assume the properties as well, which is likely to have an undesired effect as shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Possible Undesired Style Application</title>
</head>
<style type="text/css">
  <!-- input { background-color: #00FFFF;
    border: solid 3px red;
  }
  -->
</style>
<body>
  <p> Possible Undesired Style Application </p>
  <form method="get" action="mailto:markl2@cs.ucf.edu">
    <p>Name: <input type="text" id="name" style="background-
color:yellow; border:solid 2px red" /></p>
    <p><input type="radio" id="gf" value="female" />Female
  </p>
    <p><input type="radio" id="gm" value="male" />Male </p>
  </form>
</body>
</html>
```



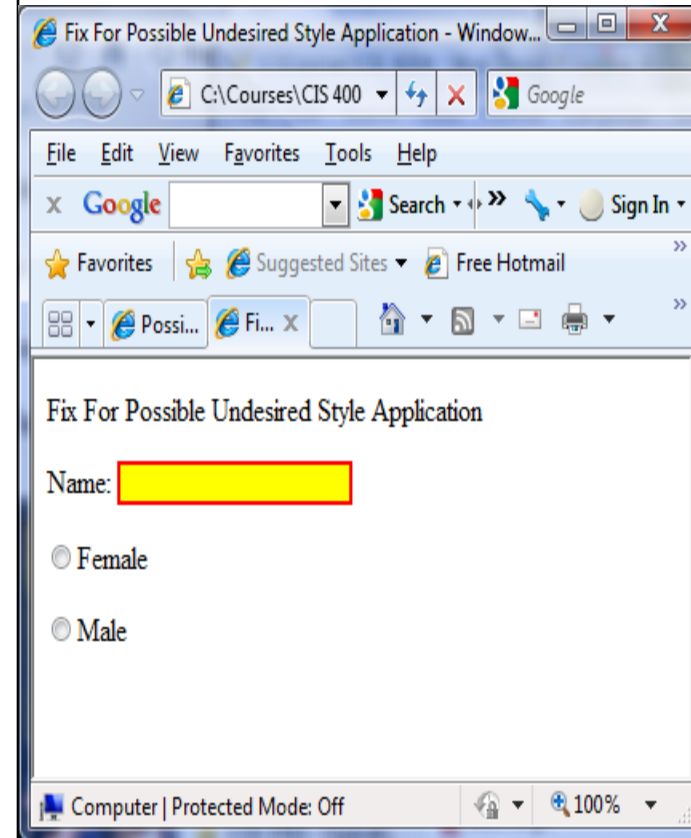
CAUTION: Undesired Style Application

- Use classes to prevent the effect shown on the previous page as illustrated here:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Fix For Possible Undesired Style Application</title>
</head>
<style type="text/css">
  <!-- .inputboxstyle { background-color: #00FFFF;
        border: solid 3px red;
        }
  -->
</style>

<body>
  <p> Fix For Possible Undesired Style Application </p>
  <form method="get" action="mailto:markl2@cs.ucf.edu">
    <p>Name: <input class="inputboxstyle" type="text"
id="name" style="background-color:yellow; border:solid 2px
red" /></p>
    <p><input type="radio" id="gf" value="female" />Female
</p>
    <p><input type="radio" id="gm" value="male" />Male </p>
  </form>
</body>
</html>
```

Define a class that applies only to certain elements.



Additional Form Controls - `<button>`

- In addition to the `<input>` form element, there are three other form elements: `<textarea>`, `<select>` which we've already seen, but additionally there is the `<button>` element.
- A button you create using the `button` element is similar to the buttons you create with the `input` element, except that the content included between the opening and closing `button` tags appears on the button.
- You can create three different types of buttons using the `button` element: `submit`, `reset`, and `button`.
- The `<button>` tag is used to create buttons .



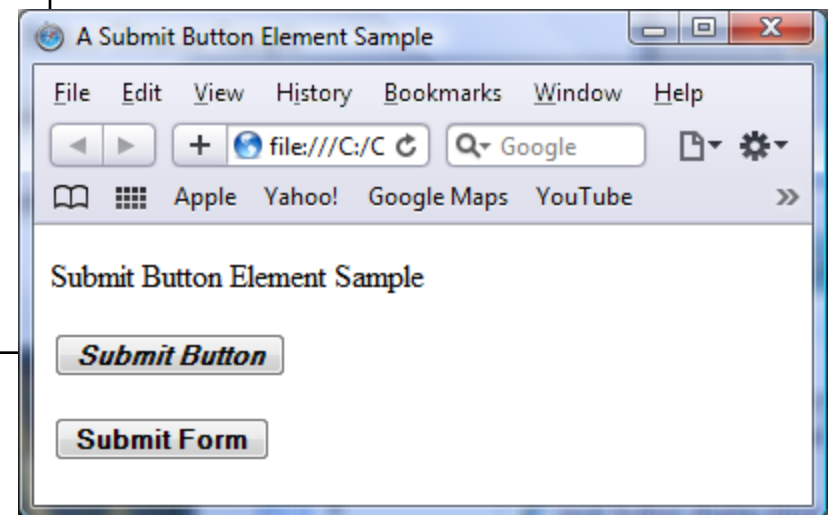
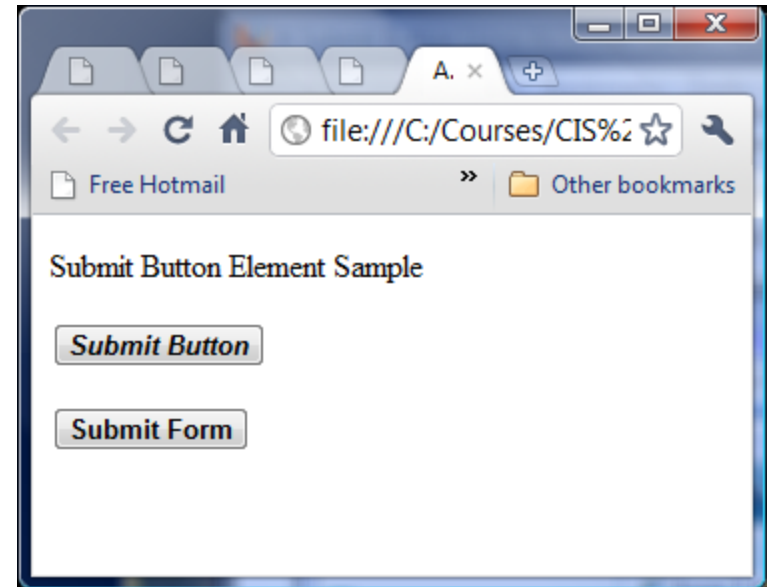
Attributes For <button> Elements

Common Attributes	Values	Usage
type	"submit", "reset", "button"	Configures the button type.
id	Alphanumeric, no spaces, begins with a letter	Provides a unique identifier for the form element.
value	Text or numeric characters	Assigns a value to the radio button that is triggered when the radio button is checked. This should be a unique value for each radio button in a group. This value can be accessed by client-side and server-side processing.



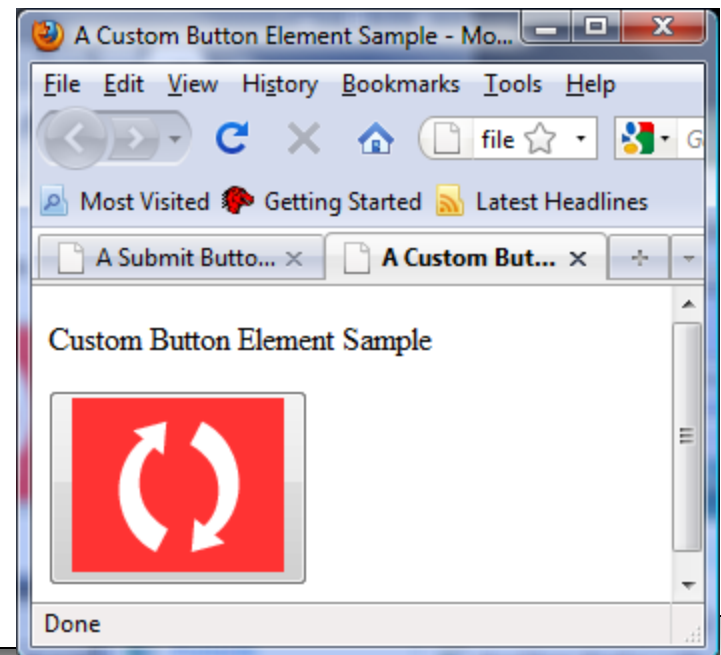
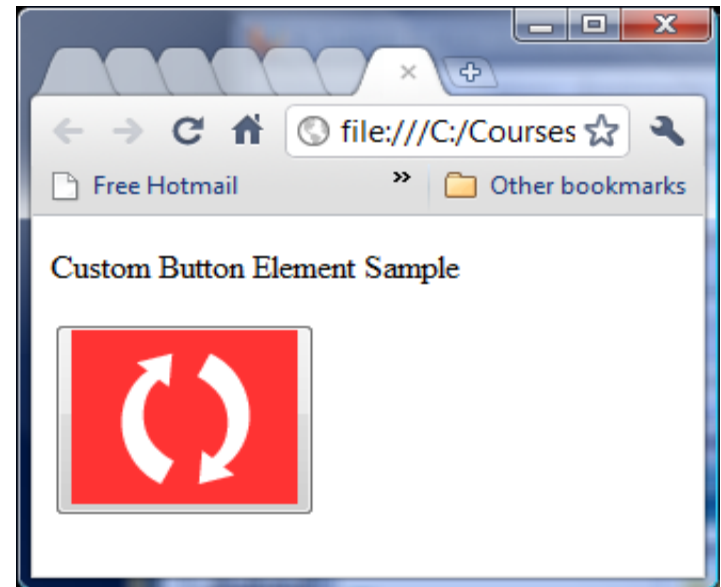
Using A <button> Element

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Submit Button Element Sample</title>
</head>
<body>
<p> Submit Button Element Sample </p>
<form method="get"
action="mailto:mark1@cs.ucf.edu">
<p><button type="submit" id="submit"
value="submit information"><b><i>Submit
Button</i></b></button></p>
<p><button type="submit" id="submit"
value="submit information"><b>Submit
Form</b></button></p>
</form>
</body>
</html>
```



Using A <button> Element

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Custom Button Element Sample</title>
</head>
<body>
<p> Custom Button Element Sample </p>
<form method="get"
action="mailto:markl@cs.ucf.edu">
<p><button type="button" id="recycle"
value="recycleform"></button></p>
</form>
</body>
</html>
```



Grouping Controls With `fieldset` and `legend`

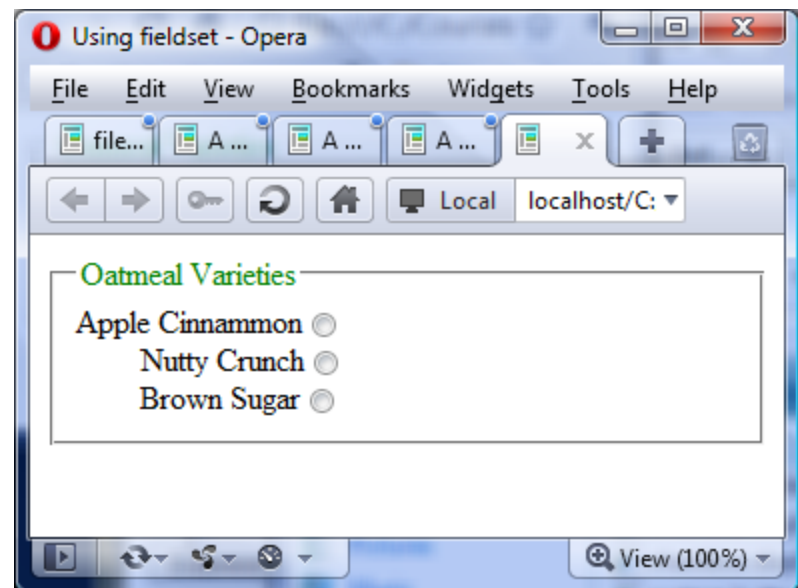
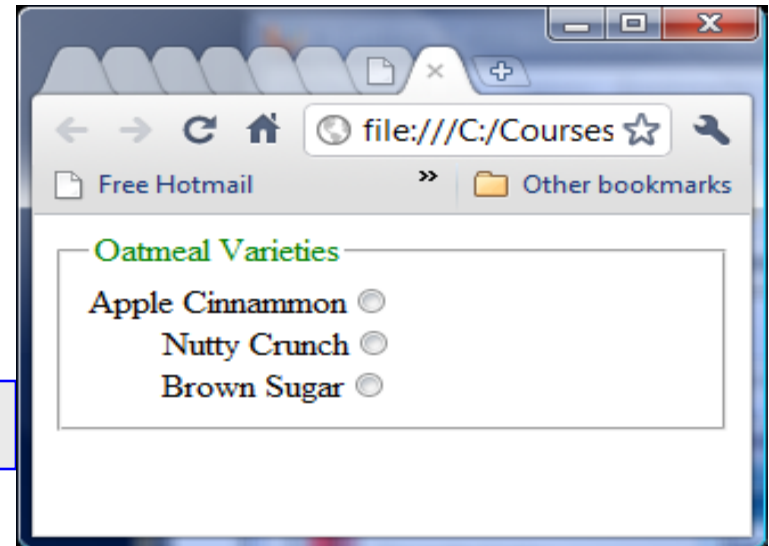
- The `fieldset` element organizes form controls into groupings that are rendered by the Web browser.
- The `legend` element displays a caption for the `fieldset` element.
- To create a `fieldset` element, start with the opening `<fieldset>` tag followed by the `<legend>` tag.



Using <fieldset> And <legend> Elements

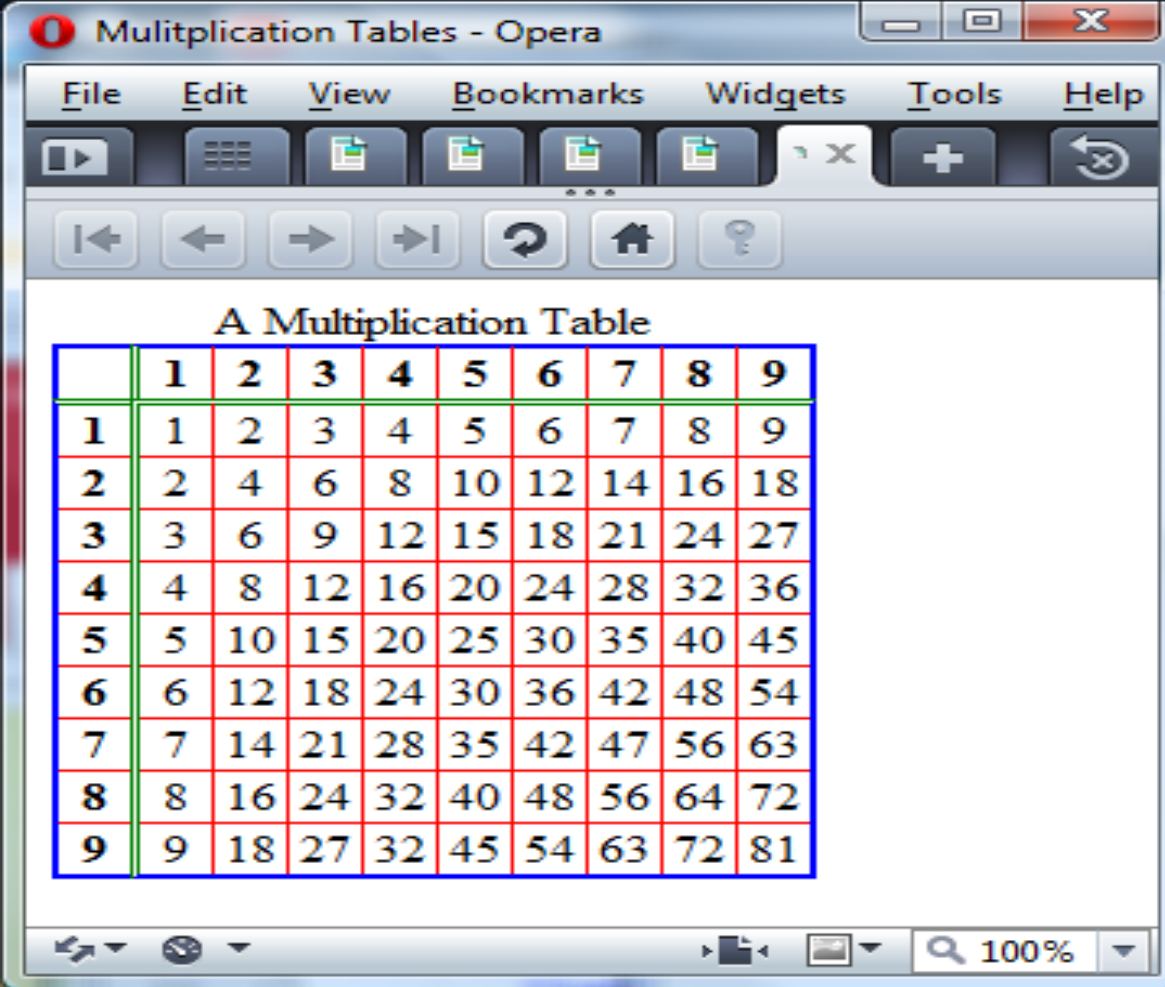
```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title> Using fieldset </title>
</head>
<body>
  <fieldset>
    <!-- <legend style="color:green">Oatmeal
Varieties</legend> -->
    <legend> Oatmeal Varieties</legend>
    <label>Apple Cinnammon<input type="radio"
id="applecinnammon" /></label><br />
    <label>&nbsp; &nbsp; &nbsp; &nbsp; Nutty
Crunch<input type="radio" name="rb"
id="nuttycrunch" /></label><br />
    <label>&nbsp; &nbsp; &nbsp; &nbsp; Brown
Sugar<input type="radio" name="rb" id="brownsugar"
/></label><br />
  </fieldset>
</body>
</html>
```

Use this to override
default browser color.



Tables: Practice Problems

1. Create a table that looks like the following:



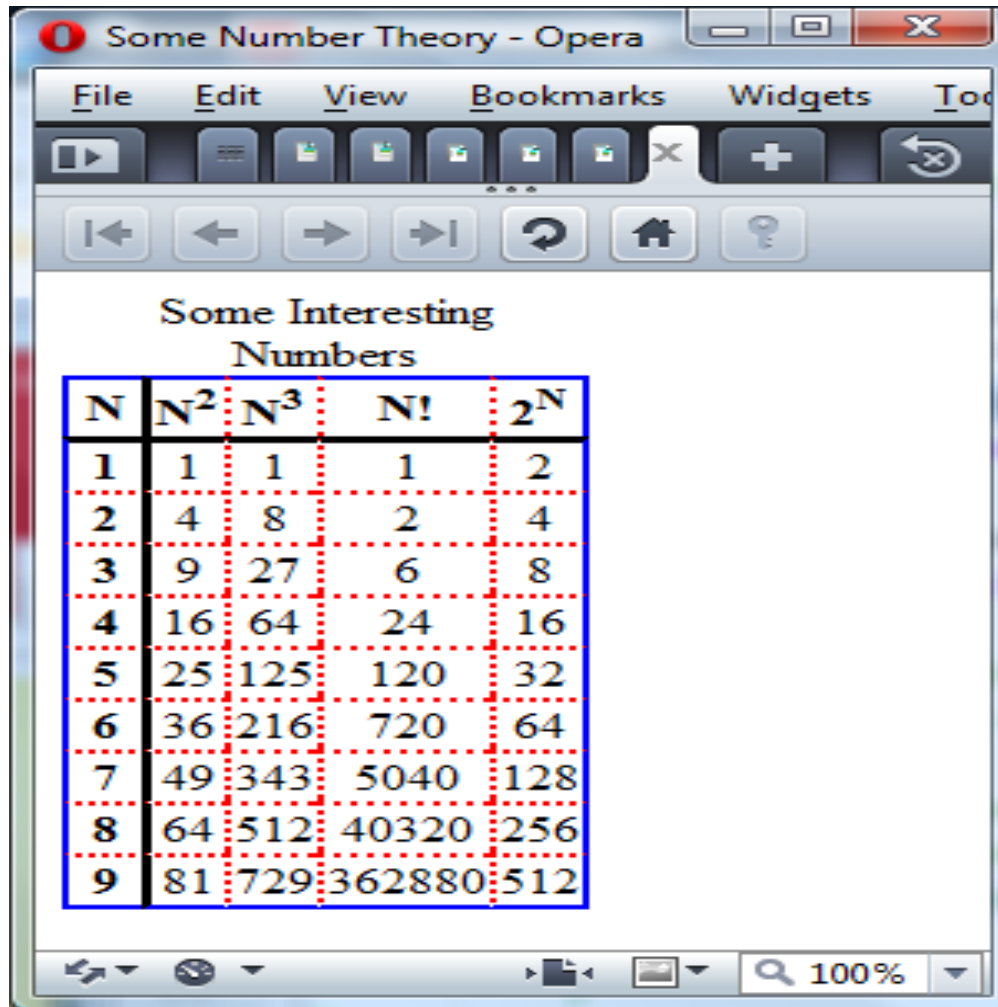
The screenshot shows a web browser window titled "Multiplication Tables - Opera". The browser's address bar is empty, and the page content displays a multiplication table. The table has a title "A Multiplication Table" and consists of a 9x9 grid of numbers. The first row and first column are highlighted with a green border, and the rest of the grid is highlighted with a red border. The numbers in the grid represent the product of the row and column indices.

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	47	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	32	45	54	63	72	81



Tables: Practice Problems

2. Create a table that looks like the following:



The screenshot shows a web browser window titled "Some Number Theory - Opera". The browser's address bar is empty. The page content is titled "Some Interesting Numbers" and contains a table with 5 columns and 10 rows. The columns are labeled N , N^2 , N^3 , $N!$, and 2^N . The rows contain numerical values for N from 1 to 9. The table is styled with a blue border and red dashed grid lines.

N	N^2	N^3	$N!$	2^N
1	1	1	1	2
2	4	8	2	4
3	9	27	6	8
4	16	64	24	16
5	25	125	120	32
6	36	216	720	64
7	49	343	5040	128
8	64	512	40320	256
9	81	729	362880	512



Forms: Practice Problems

1. Create the form as shown on page 70.
2. Modify the form from problem 1 so that it uses the `<fieldset>` tag to group the check boxes together as well as the radio boxes. It should look like the form on page 71.
3. Create the XHTML document that would produce the form shown on page 72.



Music Survey

Name:

E-mail:

Select Your Favorite Types Of Music

- Pop
- Rock
- Hip-Hop
- Classical
- Country
- Folk
- Other

Select how often you purchase music CDs:

- Daily
- Weekly
- Monthly
- A few times each year
- Never

Comments:



Music Survey

Name:

E-mail:

Select Your Favorite Types Of Music

- Pop
- Rock
- Hip-Hop
- Classical
- Country
- Folk
- Other

Select how often you purchase music CDs:

- Daily
- Weekly
- Monthly
- A few times each year
- Never

Comments:



Please Complete Our Survey

First Name: MI: Last Name:
City: State: Zip code:

Choose Your Favorite Sport

- Cycling
- Softball
- Formula One
- Table Tennis

Choose Your Class

- Freshman
- Sophomore
- Junior
- Senior

Choose Your Favorite Food

Pizza
Mexican
Chinese
Burgers and Fries

Please let us know your comments on our site --

