

1 HTML AND CSS IN BRIEF

CST272—Relational Database Programming

2 HTML TAGS

- HTML uses formatting *codes* called tags that instruct the browser how to display the document elements
- Tags are enclosed in angle brackets (<...>) ...
- Most tags are two-sided ...
 - First tag tells browser to turn on feature
 - Second tag instructs browser to turn it off
 - E.g. begins boldface, turns it off

3 SECTIONS OF A WEB DOCUMENT

- Web documents in HTML usually are divided into two main sections:
 - head: defines the title of the page, stylesheet formatting, information about the page to help search engines find it, etc.
 - body: specifies the content (visual elements) of the web page

4 HTML BASIC PAGE OUTLINE

```
<html>
```

```
<head>
```

```
  heading elements
```

```
</head>
```

```
<body>
```

```
  body elements
```

```
</body>
```

```
</html>
```

5 THE <HTML> TAG

- The <html> tag encloses all the code inside the entire HTML document
- Identifies file to the browser software as one containing HTML code, e.g.
 - Microsoft Internet Explorer
 - Firefox
- Closing tag is </html>

6 THE <!DOCTYPE> DECLARATION

- The <!DOCTYPE> declaration is an instruction to the browser about what version of HTML was used in writing the document
 - Not an actual HTML tag
- In an HTML document it must come before the <html> tag

- The following says that the type is HTML5
 - `<!DOCTYPE HTML>`

7 THE `<HEAD>` TAG

- The `<head>` tag encloses the heading elements of an HTML file, e.g.
 - `<title>` (the title bar element)
 - `<style>` (links to stylesheet files)
- Except for `<title>` the other elements are *not visible* to the viewer
- Closing tag is `</head>`

8 THE `<TITLE>` TAG

- The `<title>` tag defines the text displayed in tabs within the browser when the page is displayed
- Coded inside the `<head> ... </head>` block
- A *required* element
- Closing tag is `</title>`

11 THE `<BODY>` TAG

- The `<body>` tag encloses body elements of HTML file ...
 - The visual elements that *appear* on the web page
- Format:
 - `<body>`
- Closing tag is `</body>`

12 THE `<H1>` THROUGH `<H6>` (HEADING) TAGS

- The `<h1>` through `<h6>` tags indicate a heading line and its level ...
 - Largest font size is `<h1>`, smallest is `<h6>`
 - Text displayed in bold font style
- Used to *organize* a page into sections
- Automatic *double spacing* (carriage return/line feed) is implemented after the heading
- Closing tags are `</h1>` through `</h6>`

14 THE `<P>` (PARAGRAPH) TAG

- The `<p>` tag divides text into paragraphs
- Automatic *double spacing* (carriage return/line feed) is implemented after the paragraph
- Closing tag is `</p>`

16 THE `
` (BREAK) TAG

- The `
` tag forces a new line (*break*) in the text ...
 - Line feeds typed in the text editor are not recognized by the browser
- There is no closing tag for `
` so it should include a slash (`/`) meaning the “end” symbol *inside* the tag itself, e.g.

```
<br />
```

17 **ADDING COMMENTS**

- Text that displays in the HTML editor, but not in the browser window
- Useful for describing what developer intended when using a specific tag
- *Beginning* comment symbol is <!--
- *End* of comment symbol is -->
- Example:

```
<!--This text is a comment-->
```

18 **THE (IMAGE) TAG (PAGE 1)**

- An tag to specifies an image (graphic) file to be displayed on page
- There is no closing tag for so it should include a slash (/) meaning the “end” symbol *inside* the tag itself
- Format:

```

```
- Example:

```

```

19 **THE (IMAGE) TAG (PAGE 2)**

- In ASP.NET it is possible to drag an image file from “Solution Explorer” into the web document to automatically create the tag

20 **ATTRIBUTES**

- Many tags include additional data (properties) which modify how the element will render
- Values are assigned to these attributes
- Format:

```
<tagName attribute="value" [, ...]>
```
- Example:

```

```

21 **THE SRC ATTRIBUTE**

- For an tag , the src attribute is used in an to name an image (graphic) file to be displayed on page
- *Path* required if file is not located in same folder as the HTML document
- Format:

```

```
- Example:

```

```

22 **THE ALT ATTRIBUTE**

- For an `` tag, the `alt` attribute displays text that will appear instead *if the image does not download* (“a broken image”)

- Format:

```

```

- Example:

```

```

23 THE WIDTH AND HEIGHT ATTRIBUTES (PAGE 1)

- Sets the image’s size displayed in width and height (measured in pixels)
- If browser knows *how much room to allow*, it can fill in the rest of the text as the image continues to load
- It is better to *avoid* using width and height to make the image *smaller* ...
 - Create smaller image using graphics software
 - Will load faster and look better

24 THE WIDTH AND HEIGHT ATTRIBUTES (PAGE 2)

- Format:

```

```

- Example:

```

```

26 THE `<HR>` (HORIZONTAL RULE) TAG (PAGE 1)

- The `<hr>` (horizontal rule) tag displays a horizontal *line* across page
- Attributes that can be formatted are color, width (default: 100%) and size (thickness—default: 1)

- Format:

```
<hr />
```

27 THE `<HR>` (HORIZONTAL RULE) TAG (PAGE 2)

- There is no closing tag for `<hr>` so it should include a slash (`/`) meaning the “end” symbol *inside* the tag itself, e.g.

```
<hr />
```

29 CREATING CASCADING STYLE SHEETS (CSS)

- Stylesheets store information on how to *present* the website
- Allows the *content* to be separated from the *formatting*
- ASP.NET’s Style Builder is a graphical user interface for building CSS files

30 CSS OVERVIEW

- A style rule is the information that is applied to a single HTML tag, or a group of tags

- Syntax for the rules vary:
 - Inline rules apply to a *single tag*
 - Embedded rules apply to all elements within a *single web page*
 - External rules apply to all elements within *multiple web pages*

31 **INLINE STYLES** **(PAGE 1)**

- Placed inside the tag as a *style attribute*
 - *Join* together the property and value pairs with colons (:)
 - *Separate* the pairs using semicolons (;)
- Format:


```
<tagName style="property1:value1; property2:value2; ... " >
  Content goes here
</tagname>
```
- Example


```
<h1 style="color:green;">Welcome to Tara Store!</h1>
```

32 **INLINE STYLES** **(PAGE 2)**

- To create an inline style:
 1. Select (click on) the object in either "Source" or "Design" view
 2. Select "Style" in the "Properties" window
 3. Click the build button [...]
 4. The "Style" dialog window has formatting Categories based on the element(s) that are being formatted
 - Select the one(s) to be edited and modify those
 5. Click the <OK> button and the rules are applied to the selected element in a "style" attribute

34 **EMBEDDED STYLESHEETS** **(PAGE 1)**

- An embedded stylesheet is placed in the <head> section
 - Style content is placed within a <style>...</style> block
- Uses the same coding format as in the style attribute
 - E.g. *property1: value1; property2: value2; ...*
- Style elements will apply to *every* matching tag (or other selector) in the web page

35 **EMBEDDED STYLESHEETS** **(PAGE 2)**

- Format:


```
<style>
  selector1 {
    property1: value1;
```

```

        property2: value2;
    ...
}
selector2 {
    ...
}
</style>

```

36 **EMBEDDED STYLESHEETS (PAGE 3)**

- Example:

```

<html>
<head>
  <title>Gerry's Homepage</title>
  <style>
    h2 {
      font-family: Verdana;
      color: #FF00FF;
    }
  </style>
</head>
<body>
  <h2>Gerry's Homepage</h2>
</body>
</html>

```

37 **ADDING AN EMBEDDED STYLE RULE (PAGE 1)**

- To add a style rule to an embedded stylesheet:
 1. Move to the web document in "Design" view
 2. Select Format from the menu bar
 3. Select New Style... from Format menu ("New Style" window appears)

38 **ADDING AN EMBEDDED STYLE RULE (PAGE 2)**

- To add a style rule to an embedded stylesheet: (*con.*):
 4. In the "New Style" dialog window
 - a. Select or type the "Selector:", e.g. h2
 - b. Select "Current Page" for "Define in:"

39 **ADDING AN EMBEDDED STYLE RULE (PAGE 3)**

- To add a style rule to an embedded stylesheet: (*con.*):

5. The “New Style” dialog window has formatting Categories based on the element(s) being formatted; select the one(s) to be edited and modify
6. Click the <OK> button and the rules are added to the embedded stylesheet for that selector

41 **ADDING AN EMBEDDED STYLE RULE (PAGE 4)**

- More than one element can be included as the selector in a comma-delimited list to create the rule for multiple HTML elements
 - E.g. h2, p which means for both h2's and p's
- HTML elements can be combined to create a hierarchy,
 - E.g. ul li which means li's only for ul's (not ol's)

44 **UNORDERED LISTS**

- Also called a *bulleted* list
- A block contained within the HTML tags ... (*unordered list*)
- The tag for each item is (*list item*)
- The unordered list automatically *indents* each item in list to second level

45 **FORMAT OF THE TAG**

- Format:


```
<ul style="list-style-type: disc | circle | square">
  <li>text</li>
  <li> ...
</ul>
```

46 **UNORDERED LIST EXAMPLE**

```
<ul>
  <li>Delete all META tags</li>
  <li>Change the title to EXAM 1</li>
  <li>Add the background image</li>
</ul>
```

47 **ORDERED LISTS**

- Also called a *numbered* list
- A block contained within the HTML tags ... (*ordered list*)
- The tag for each item is (*list item*)
- The ordered list automatically *indents* each item in list to second level

48 **FORMAT OF THE TAG**

- Format:


```
<ol style="list-style-type: decimal | lower-alpha | lower-roman | upper-alpha |
  upper-roman">
  <li>text</li>
```

```
<li> ...
</ol>
```

49 ORDERED LIST EXAMPLE

```
<ol>
  <li>Click the Numbering button once</li>
  <li>Type text and press Enter key</li>
  <li>Repeat the above</li>
  <li>At the end preview document</li>
</ol>
```

51 THE <A> (ANCHOR) TAG

- The href (hyperlink reference) attribute of <a> (anchor) tag links to another web page

- Format:

```
<a href="http://URL">text/object</a>
```

- Example:

```
<a href="http://www.sunysuffolk.edu">SCCC home page</a>
```

- Appears in a web page as: [SCCC home page](#)

52 HYPERLINK TO E-MAIL ADDRESS

- Uses mailto: reference in href (hyperlink reference) option of the <a> (anchor) tag

- Launches e-mail software and addresses the message

- Format:

```
<a href="mailto:e-mailAddress">text/object</a>
```

- Example:

```
<a href="mailto:struckc@sunysuffolk.edu">
```

```
  Send e-mail to Prof. Struck</a>
```

- Appears in a web page as: [Send e-mail to Prof. Struck](#)

54 CREATING A TABLE (PAGE 1)

- A table is a block inserted within the tags:

```
<table> ... </table>
```

- Rows are inserted within the <table> using:

```
<tr> ... </tr>
```

- Cells (table data) are inserted into the <row>'s using:

```
<td> ... </td>
```

55 CREATING A TABLE (PAGE 2)

- The border property determines if there is a border around the table and individual cells

- Examples:


```
<table border="1"> (there is a border)
<table border="0"> (there is no border—default)
```

- There also is style property for border

56 A TABLE EXAMPLE

```
<table border="1">
  <tr>
    <td>Some data</td>
    <td>More data</td>
    <td>End of line</td>
  </tr>
  <tr>
    <td>Start of second row</td>
    <td>Middle of second row</td>
    <td>End of second row</td>
  </tr>
</table>
```

57 INSERTING A ROW

- Rows are inserted into a table using the `<tr>` (table row) tag

- Format:

```
<table>
  <tr>
    ... cells_within_the_row ...
  </tr>
</table>
```

58 CREATING CELLS AND ENTERING DATA

- Content is inserted into cells within a table using the `<td>` (table data) tag
- Format for a single cell:

```
<td>content within the cell</td>
```

- Example:

```
<table>
  <tr>
    <td>Some data</td>
    <td>More data</td>
  </tr>
</table>
```

59 CREATING A TABLE EXAMPLE

```
<table border="1">
  <tr>
```

```

    <td>Some data</td>
    <td>More data</td>
    <td>End of line</td>
</tr>
<tr>
    <td>Start of second row</td>
    <td>Middle of second row</td>
    <td>End of second row</td>
</tr>
</table>

```

60 **ADDING TABLE HEADERS** (PAGE 1)

- Table headers are titles displayed in bold and centered *above* each of the columns of the table
- The <th> tags usually are used within the first <tr> (row) of the table to add the headings
 - Design of a table usually includes as many <th> headings as there are columns
- Example:


```
<th>Monday</th>
```

61 **ADDING TABLE HEADERS** (PAGE 2)

```


<table border="1">
  <tr>
    <th>Heading 1</th>
    <th>Heading 2</th>
    <th>Heading 3</th>
  </tr>
  <tr>
    <td>Some data</td>
    <td>More data</td>
    <td>End of line</td>
  </tr>
  <tr>
    <td>Start of second row</td>
    <td>Middle of second row</td>
    <td>End of second row</td>
  </tr>
</table>

```

62 **ADDING TABLE CAPTIONS**

(PAGE 1)


- The <caption> tag places a center-aligned “title” above the table
- There can be *only one* caption per table, and the <caption> tag must be inserted right after the <table> tag

63  **ADDING TABLE CAPTIONS**
(PAGE 2)


- Format:

```
<caption>caption text</caption>
```
- Example:


```
<table border="1">
  <caption>Lemonade Stand</caption>
  ...
</table>
```

64  **EXTERNAL STYLESHEETS**
(PAGE 1)

- An external stylesheet is created as a *separate document file* with the extension .css
- Promotes uniform style for an *entire* website
- Uses same coding format found in the style attribute and the <style> block for every element in a linked stylesheet

65  **EXTERNAL STYLESHEETS**
(PAGE 2)

- The <link> tag in the <head> section tells the browser to look at the external stylesheet file for formatting elements
- This tag is placed into every web page that implements the stylesheet file
- The <link> tag attaches a stylesheet to a web page


66  **EXTERNAL STYLESHEETS**
(PAGE 3)

- Format:

```
<link href="filename.css" type="text/css" rel="stylesheet" />
```

 - href—location of file (relative or absolute URL)
 - type—the file is a “text/css” MIME type
 - rel—the file is a CSS stylesheet (relationship)
- Example:

```
<link href="style.css" type="text/css" rel="stylesheet" />
```

67  **EXTERNAL STYLESHEETS**
(PAGE 4)

- In ASP.NET a <link> tag can be inserted by *dragging* a stylesheet file from the “Solution Explorer” window into a Web Form
- Either in Design or HTML view

68 **STYLESHEET RULES AND HIERARCHY**

- CSS may contain conflicting styles when there are *different rules* for the same elements within inline, embedded and/or external
- In general:
 - Inline takes precedence over embedded and external
 - Embedded takes precedence over external

69 **COMMENTS IN STYLESHEETS**

- The characters `/* */` (like Java) can be added to embedded or external stylesheets to insert internal documentation (comments)

- Example:

```
/* Stylesheet1.css
   Created By: Prof. Struck
   Stylesheet is used to format main menu */
/* Corporate logo */
h1 {color:green;}
/* Red heading */
h2 {color:red;}
/* Blue heading */
h3 {color:blue;}
```

70 **CREATING AN EXTERNAL STYLESHEET (PAGE 1)**

- To create a *new* stylesheet:
 1. Select Project from the menu bar
 2. Select Add New Item... from the Project menu
 3. Scroll down in the Templates: window and select "Style Sheet"
 4. Type a Name: for the new file (".css" extension is added automatically) and click the <Open> button

71 **CREATING AN EXTERNAL STYLESHEET (PAGE 2)**

- Then drag stylesheet document from "Solution Explorer" into each web document to "link" it
 - Either before, during or after creating CSS formatting elements in stylesheet

72 **EXTERNAL STYLESHEET EXAMPLE (PAGE 1)**

```
body {
```

```
}
```

```
ul li {
```

```
  color: #008000;
```

```
font-family: Verdana;
font-size: 10pt;
list-style-image: url('images/bullet.gif');
}
```

73 **EXTERNAL STYLESHEET EXAMPLE
(PAGE 2)**

```
table {
border-style: solid;
border-color: #008000;
}
```

```
tr {
text-align: center;
}
```

```
.TSHeadings {
font-family: Elephant;
font-variant: small-caps;
font-size: 10pt;
}
```

74 **EXTERNAL STYLESHEET EXAMPLE
(PAGE 3)**

```
.txtBox {
font-weight: bold;
color: #008000;
font-family: Verdana;
font-size: 10pt;
border-style: solid;
border-color: #008000;
}
```

75 **EXTERNAL STYLESHEET EXAMPLE
(PAGE 4)**

```
#ButtonSubmit {
font-weight: bold;
color: white;
font-family: Verdana;
font-size: 10pt;
font-variant: small-caps;
background-color: #008000;
```

```
border-color: #FFFF00;
border-style: ridge;
}
```

77 ADDING A STYLE RULE (PAGE 2)

- To add a new style rule to a stylesheet (*con.*):
 4. In the “New Style” dialog window
 - a. Select or type the “Selector:”, e.g. h2
 - b. Select “Existing style sheet” for “Define in:”
 - c. Click <Browse> button and select filename of the stylesheet for “URL” (unless filename is already there)

78 ADDING A STYLE RULE (PAGE 3)

- To add a new style rule to a stylesheet (*con.*):
 5. The “New Style” dialog window has formatting Categories based on the element(s) that are being formatted; select those to be edited and modify
 6. Click the <OK> button and the rules are added to the stylesheet for that selector

84 CLASSES (PAGE 1)

- A class name can be used to format a *group* of different tags or a subgroup of a specific tag
- In the CSS document, prefix the name of the class with a period (.)

```
.SelCat {
    color: red;
}
.BlueHead {
    color: blue;
}
```

85 CLASSES (PAGE 2)

- Apply the class with attribute name class and the class name (see below):

```
<body>
<h1 class="BlueHead">Welcome to Tara Store!</h1>
<h2>Product Listing:</h2>
<ul>
  <li class="SelCat">Gifts</li>
  <li>Jewelry</li>
  <li>China & Crystal</li>
```

```

    <li>Clothing</li>
    <li>Books, Music, & Videos</li>
</ul>

```

86 CLASSES (PAGE 3)

- The headings on both this and the previous page would appear in blue:

```

<h3 class="BlueHead">About Tara Store</h3>
<ul>
  <li>What's New</li>
  <li>Current Sales</li>
  <li>Location</li>
  <li>Contact Us</li>
  <li>Members Only</li>
</ul>
</body></html>

```

87 ADDING STYLE RULE FOR A CLASS (PAGE 1)

- To add a new style rule for a class name to stylesheet also use the “New Style” dialog window and then:
 1. In the “New Style” dialog window
 - a. Type the class name as the “Selector:”, e.g. .TSHeader
 - b. Select “Existing style sheet” for “Define in:”
 - c. Click <Browse> button and select filename of the stylesheet for “URL” (unless filename is already there)

88 ADDING STYLE RULE FOR A CLASS (PAGE 2)

- To add a new style rule for a class name to stylesheet also use the “New Style” dialog window (*con.*):
 2. The “New Style” dialog window has formatting Categories based on the element(s) that are being formatted; select those to be edited and modify
 3. Click <OK> button and rules are added to stylesheet for that class

91 THE ID ATTRIBUTE (PAGE 1)

- The id attribute in HTML is used to provide each tag with a *unique* name
 - The same is true for ASP.NET server controls
- Format:


```
<htmlElement id="idName">
```

 - The *htmlElement* is a type of tag, e.g. <h2>, etc.
- Example:

```
<input id="TSButton" type="Button" value="Send" />
```

92 THE ID ATTRIBUTE (PAGE 2)

- To apply CSS formatting in the stylesheet document to an “id” attribute, place a hash character (#) before the id name

- Format:

#idSelector

- Example:

```
#TSButton {  
    color: white;  
    font-family: Verdana;  
    background-color: #008000;  
}
```

93 ADDING STYLE RULE FOR AN ID (PAGE 1)

- To add a new style rule for an id to a stylesheet also use the “New Style” dialog window and then:

1. In the “New Style” dialog window
 - a. Type the class name as the “Selector:”, e.g. #ButtonSubmit
 - b. Select “Existing style sheet” for “Define in:”
 - c. Click <Browse> button and select filename of the stylesheet for “URL” (unless filename is already there)

94 ADDING STYLE RULE FOR AN ID (PAGE 2)

- To add a new style rule for an id to a stylesheet also use the “New Style” dialog window (*con.*):
 2. The “New Style” dialog window has formatting Categories based on the element(s) that are being formatted; select those to be edited and modify
 3. Click <OK> button and rules are added to stylesheet for that selector