

HTML Tables and Forms

Dr. Michele Weigle

<http://www.cs.odu.edu/~mweigle/CS418-F12/>

Outline

- ▶ Assigned Reading
 - ▶ Chapter 4 “Using Tables to Display Data”
 - ▶ Chapter 5 “Form Elements: Letting the User Work with Data”
- ▶ HTML Tables Review
- ▶ HTML Forms Review
- ▶ Chs 4, 5 Code Example Demo/Walkthrough

HTML Tables

Syntax

- ▶ Tables are defined with the `<table>` tag
- ▶ Rows are defined with `<tr>`
- ▶ Cells are defined with `<td>`
- ▶ Headings are defined with `<th>`

HTML Tables

Cells

- ▶ Each item in a table is placed into a cell
- ▶ A cell's width specified by either
 - ▶ number of pixels
 - ▶ percentage of the table width
- ▶ If not specified, a cell's dimension is dynamically determined based on the content in the cell
 - ▶ all items in the same column are of the same width
 - ▶ the widest item determines the column width
 - ▶ an item is as narrow as it is allowed to be
- ▶ A cell's height is determined in the same way as width

HTML Table

Cell Sizing

- ▶ When improperly specified, different browsers may treat the table in different ways
 - ▶ to interpret and display in their respective best possible ways
- ▶ When either or both of the width and height are specified, resizing the font by user may cause display problems
- ▶ Unless specified, final cell dimensions are generally based on the data in the entire table

HTML Tables

Features of Table Cells

- ▶ Individual colored background
- ▶ Individual content alignment
- ▶ Allow better controlled image display in a page
 - ▶ Example: row of pictures with captions in row below

HTML Tables

Simple Example

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

<http://www.cs.odu.edu/~mweigle/cs312/html/table.html>

HTML Tables

Example with Headings

```
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

HTML Tables

Blank Cell

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td></td>
</tr>
</table>
```

- ▶ Note: Some browsers will display the empty cell differently if it contains ` `; rather than nothing (or just spaces).

HTML Tables

Attributes

- ▶ **border**
 - ▶ default is no border (or `border = "0"`)
- ▶ **cellpadding**
 - ▶ amount of space between the contents of the cell and the cell wall, default is 1
- ▶ **cellspacing**
 - ▶ defines the space between cells, default is usually 2
- ▶ **width**
 - ▶ width of table defined in pixels (`width="300"`) or percentage of the browser window (`width="75%"`)
 - ▶ if defined in percentage, resizing window resizes the table
- ▶ Attributes such as this can also be specified via CSS

<http://www.cs.odu.edu/~mweigle/cs312/html/table-attr.html>

HTML Tables

Multiple Rows/Columns

- ▶ **colspan**
 - ▶ defines a cell to be a certain number of columns across
- ▶ **rowspan**
 - ▶ defines a cell to be a certain number of rows high
- ▶ Both are defined as attributes of the cell
 - ▶ either in `<td>` or `<th>`

HTML Tables

Color, Captions, and Cell Width

- ▶ **bgcolor**
 - ▶ can be used as attribute for `<table>` or `<td>`
- ▶ **caption**
 - ▶ specified by `<caption> ... </caption>` tags inside the table
 - ▶ must right after `<table>` in XHTML
 - ▶ can use align attribute (bottom, top, left, right)
- ▶ **cell width / height**
 - ▶ attribute of `<td>` and `<th>`
 - ▶ either in pixels or percentage of table width

<http://www.cs.odu.edu/~mweigle/cs312/html/table-fancy.html>

HTML Tables

Nested Tables

- ▶ Tables within tables
- ▶ Any cell can be another table

```
<table border="1">
<tr><td>cell 1</td>
<td>cell 2</td></tr>
<tr>
<td><table><tr>
<td>subcell 1</td>
<td>subcell 2</td></tr>
<tr>
<td>subcell 3</td>
<td>subcell 4</td></tr>
</table></td>
<td>cell 3</td></tr>
</table>
```

Outline

- ▶ HTML Tables Review
- ▶ HTML Forms Review
- ▶ Chs 4, 5 Code Example Demo/ Walkthrough

HTML Forms

- ▶ Form elements are elements that allow the user to enter information
 - ▶ text fields
 - ▶ radio buttons
 - ▶ checkboxes
 - ▶ buttons
 - ▶ drop-down menus
 - ▶ textareas
- ▶ The definition and layout of a form is HTML, but a server-side script is needed to process the data provided to the forms.

HTML Forms

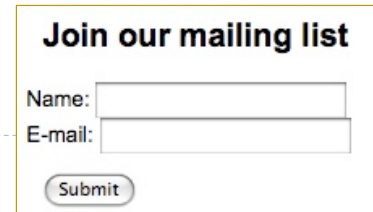
- ▶ Defined with the `<form>` tag:

```
<form action="">
<label> </label> <input />
...
<label> </label> <input />
</form>
```

- ▶ action attribute is required by XHTML.
- ▶ User input fields are defined by the `<input />` tag
 - ▶ attributes: type (type of input), name (used for referencing)
- ▶ Text label associated with an input field is defined with the `<label>` tag

HTML Forms

Example



Join our mailing list

Name:

E-mail:

```
<form action="maillist.pl" method="post">

<h2>Join our mailing list</h2>

<label>Name:</label> <input type="text"
name="realname" />

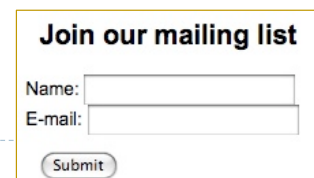
<br /><label>E-mail:</label> <input type="text"
name="email" />

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

</form>
```

HTML Forms

Forms and Actions



Join our mailing list

Name:

E-mail:

- ▶ The main attribute of a form tag is action
 - ▶ ex: `<form action="/cgi-bin/maillist.pl">`
 - ▶ action is required by XHTML
 - ▶ if no action to be taken, then `action=""` will validate
- ▶ action tells the browser where to send the data for processing
- ▶ `input type="submit"` creates the submit button
 - ▶ when pressed, the data is sent to the action defined

HTML Forms

```
<form action="maillist.pl" method="post">
```

Forms and Methods

- ▶ If the method is **get**
 - ▶ query string of the arguments is tacked onto the end of the URL (of action attribute)
 - ▶ name=value
 - ▶ "?" is separator between data-value pairs
 - ▶ URL is sent to the web server
 - ▶ should only be used when doing a search or requesting data
- ▶ If the method is **post**
 - ▶ client sends the query string directly to the server, separately from the URL
 - ▶ should be used when updating data on the server, for example, in a database

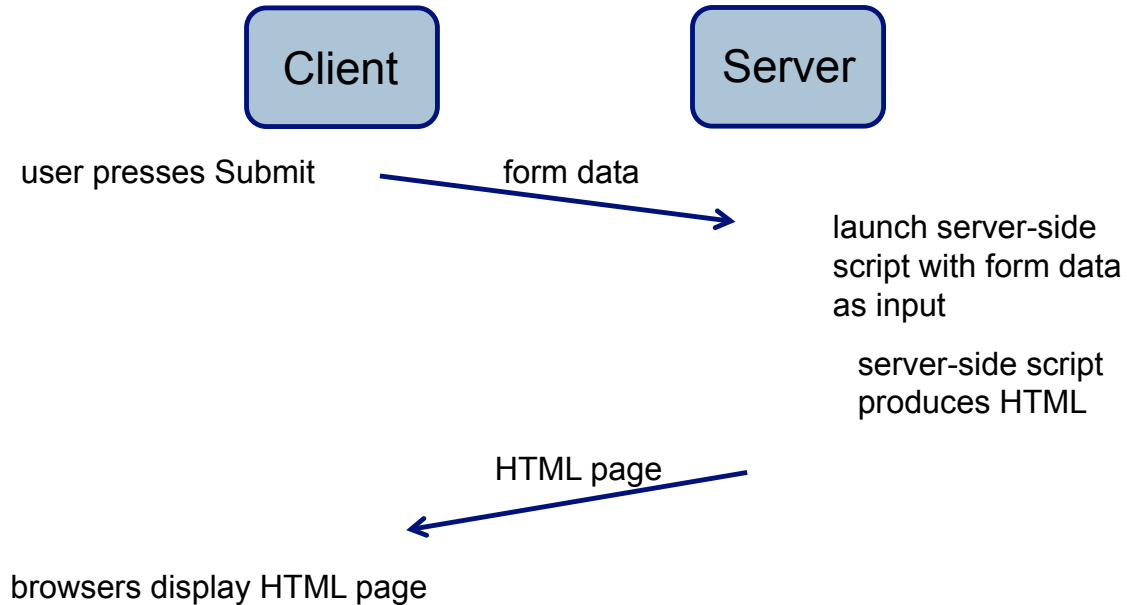
Form Processing

What Happens When Submit is Pressed?

- ▶ User presses "Submit" button
- ▶ Browser sends form data to web server
 - ▶ specifically, to server-side script defined in `<form action>`
- ▶ Web server launches the server-side script
- ▶ Server-side script executes, taking the data from the form as input
- ▶ Server-side script typically will generate a web page using HTML
- ▶ Server-side script passes the HTML page back to the web server
- ▶ Web server passes the HTML page back to the browser

Form Processing

What Happens When Submit is Pressed?



HTML Forms

Forms and Server-Side Scripts

- ▶ For the simple examples, we'll use the HTML Code Tutorial's mycgi.pl script
 - ▶ <http://www.htmlcodetutorial.com>
- ▶ Displays name=value pairs that are sent to it

HTML Forms

Form Input Types


- ▶ Text
- ▶ Submit Button
- ▶ Reset Button
- ▶ Password
- ▶ Radio Button
- ▶ Checkbox
- ▶ Non-Input Types
 - ▶ select (scrolling or drop-down list)
 - ▶ textarea

Text Type

- ▶ A one-line text entry field

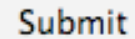
```
<input type="text" name="user" value="Donald Smith" size="30" />
```

- ▶ Attributes:
 - ▶ type
 - ▶ name of this parameter
 - ▶ value (optional) – default input value
 - ▶ size (optional) – field width
 - ▶ maxlength (optional) – limit the number of characters the user can enter
- ▶ When form is submitted, the information will be passed as user=Donald+Smith



Submit Type

- ▶ A submit button



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

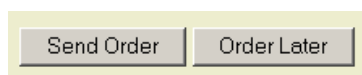
- ▶ Value indicates the text that will be placed on the button
 - ▶ if nothing given, default is "Submit Query"
- ▶ **Important:** When pressed, the form data is submitted to the script specified the form's action attribute

Submit Type

Multiple Submit Buttons

- ▶ Can have multiple submit buttons in the same form
- ▶ We can have multiple submit buttons, e.g., two labeled Send Order and Order Later, respectively.

```
<input type="submit" name="action" value="Send Order" />  
<input type="submit" name="action" value="Order Later" />
```



Submit Type

Multiple Submit Buttons

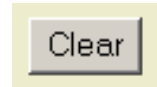
- ▶ When multiple submit buttons are used in a single form, they should have the same name but different values
- ▶ Only one submit button can be clicked/effective.
 - ▶ If the user clicked on the button labeled Send Order, then the corresponding part of the query string will be action=Send+Order

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Reset Type

- ▶ A reset button

```
<input type="reset" value="Clear" />
```

A rectangular button with a light gray background and a thin black border. The word "Clear" is centered on the button in a black, sans-serif font.
- ▶ Value indicates the text that will be placed on the button
 - ▶ if nothing given, default is "Reset"
- ▶ When pressed, the all field data and selections in the form are reset back to their original, default values

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Hidden Field

- ▶ Used to pass some value, not given in any current input fields, to the called procedure.

```
<input type="hidden" name="to" value="weigle" />
```

- ▶ In the query string, this field and value pair are passed as
 - ▶ to=weigle
- ▶ But, nothing is shown in the document text or form

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Password Type

- ▶ A one-line password entry field



```
<input type="password" name="passwd" value="xyzzzy" size="10" />
```

- ▶ All characters, default or user input, in the password field are shown as asterisks or dots.
- ▶ When form is submitted, the information will be passed as
 - ▶ passwd=xyzzzy
 - ▶ no encryption is performed (plain-text)

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Radio Button Type

- ▶ A group of radio buttons
 - ▶ Similar to checkboxes, but the user can select only one out of a group

☒ Small ☐ Medium ☐ Large

```
<input type="radio" name="size" value="small"
checked="checked" />Small
<input type="radio" name="size" value="medium" /> Medium
<input type="radio" name="size" value="large" />Large
```

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Checkbox Type

- ▶ A group of checkboxes
 - ▶ Used to select multiple items.

☐ Engine ☒ Tire ☐ Seat

```
<input type="checkbox" name="items" value="engine" />Engine
<input type="checkbox" name="items" value="tire"
checked="checked" /> Tire
<input type="checkbox" name="items" value="seat" /> Seat
```

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Select Tag

- ▶ A drop-down or scrolling list

- ▶ drop down - `<select name="cars">`
- ▶ scrolling - `<select name="favorites" size="4" multiple="multiple">`



- ▶ Each option in the list is surrounded by `<option>...</option>` tags

- ▶ ex: `<option>jogging</option>`
- ▶ for default selection, use selected attribute on option tag
 - ▶ ex: `<option selected="selected">swimming</option>`

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>



Button Type

- ▶ A push button

```
<input type="button" value="Press Me!" />
```

- ▶ Used to implement client-side scripts

- ▶ e.g., JavaScript
- ▶ nothing is sent to the server

- ▶ Example with simple JavaScript

```
<input type="button" value="Click!"  
onclick="javascript:alert('Clicked!');" />
```

<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Textarea Tag

- ▶ textarea tag, not an input tag
 - ▶ For defining a large input text area, not just a field of a single line, use textarea tag.

```
<textarea name="longtext" rows="5" cols="60">  
</textarea>
```



<http://www.cs.odu.edu/~mweigle/cs312/forms/form.html>

Communicating with Scripts Via URLs

- ▶ Scripts may or may not require arguments from users.
- ▶ The arguments are called a query string and may be appended at the end of a URL with the question mark "?" leading it.
- ▶ If argument has blank space
 - ▶ use "+" or "%20"
- ▶ If there are two or more name/value pairs
 - ▶ use "&" to delimit

Communicating with Scripts Via URLs

Examples

- ▶ One argument with parameter and value
 - ▶ <http://www.google.com/search?q=titanic>
- ▶ Argument value has blank space
 - ▶ <http://www.google.com/search?q=john+smith>
- ▶ Two or more parameters, using ‘&’ to link pairs
 - ▶ <http://finance.yahoo.com/q/bc?s=AAPL&t=2y>

Server-Side Actions

Details

- ▶ When a web server receives a server-side (CGI) request:
- ▶ It creates a set of environment variables containing information about
 - ▶ the server itself
 - ▶ the remote browser
 - ▶ the current request, including QUERY_STRING
- ▶ It calls the corresponding script with any arguments in the environment variable QUERY_STRING.

Server-Side Actions

Details

- ▶ The script picks up any information it wants from the environment variables, particularly the arguments from QUERY_STRING
 - ▶ i.e. the parameters with corresponding values
 - ▶ many programming languages provide tools for easy picking of parameter values by procedures
- ▶ The script then executes its own instructions
- ▶ The output by the script, typically a HTML page, is sent back to the client by the server

Outline

- ▶ HTML Tables
- ▶ HTML Forms
- ▶ Chs 4, 5 Code Example Demo/
Walkthrough

Up Next: Project 1 Status Reports, Forms + MySQL

Assigned Reading: Ch 6

Demo/Walkthrough Time

- ▶ Examples from Chapters 4, 5

- ▶ Ch 4 <http://mweigle418.cs.odu.edu/~mweigle/textbook/ch04.htm>
 - ▶ Creating an HTML Table
 - ▶ Populating the HTML Table
 - ▶ Master/Child Relationships
 - ▶ Relationships

- ▶ Ch 5 <https://mweigle418.cs.odu.edu/~mweigle/textbook/ch05.htm>
 - ▶ Simple Form
 - ▶ Movie Search Form
 - ▶ Simulate Search/Add Form for Movie, Actor, and Director