CGT 141/CPT 141 Lecture 8 Wk 6

Forms and Data Submission

Understanding Forms

- Purpose:
  - Predominantly used for the collection of data from the end user
  - Was designed as primary mechanism for extending HTML and giving it a utilitarian purpose
  - Limiting when uses apart from other technologies
- Results:
  - Data send from a form is composed of a raw series of name and value pairs.
    - The name aspect is the name of the form element from which the data is coming; defined in the HTML
    - The value is either the default value for the form element (defined in the HTML page) or the value the user entered or selected
    - Can also send “attachments” via file upload. However, must be some mechanism on the server (add-on server-side element) to support this.
    - The data, in its simplest form, is a continuous string of name and value pairs separated by semi-colons:
      - fname=ronald;lname=glotzbach;radchoice1=1
  - Data sent from a form can be sent in two ways:
    - GET – URL-encodes the data and attaches it to the end of the URL
      - URL-encoding encodes the data by replacing hexadecimal character representations for non-valid URL characters.
        - Spaces are converted to %20
        - Slashes are converted to %2F
      - Everything from the ? on, is called a QueryString:

```
resizeStore.html?src=atest/Images/BellTower/t_bellTower08.jpg&width=200&height=400
```

      - GET is not a secure method (as the data is shown in the URL bar)
    - POST – sends the data inside the server’s response header to the user’s browser
    - Ultimately: which method used is based on where the data is being sent and what data is being sent.
- Parts of the “form process”
  - Creation of the HTML form page
  - Validation
    - Cannot depend on user’s doing anything logical or of them following instructions
    - Validation allows you to make sure the user:
      - Entered something
      - Entered something valid
    - Typically done with client-side scripting, such as JavaScript or VBScript
      - Client-Side JavaScript, JScript, and VBScript can be seen by the user on the client machine – just view the source code.

            - Allows you to look at their code and grab things you deem useful.
            - Be careful of copyrighted material. Attribute the work to the author.
        - More advanced sites use server-side scripting as there are advantages for doing so.
            - Best examples: ASP, PHP, JSP, ASP.NET
            - Use one of the above to grab the form data, store it, manipulate it, send it elsewhere, … all unannounced to the user.
    - Source to send the data to
        - Simplest: send data to an email
        - More complex (and server dependent): server-side script
            - Connect to databases

Form Tags and Attributes

- <form>…</form>
    - Three (important) attributes:
        - Name & ID
            - Allows you to access the form and its elements via client-side scripting
            - Should be used anytime form data is being sent somewhere.
        - Action
            - Tells where to send the data.
            - Could be an email or various types of CGI
            - More often, it is another page that gathers and stores the data.
        - Method
            - How to send the data (GET or POST)
        - Enctype
            - Encoding Type
            - Defines what type of data is being sent and how it is encoded
            - If none included, assumed to be standard URL-encoding
            - Not necessary for the form to work – but must be included on forms that upload a file to the server.
- Labels and Controls
    - <label>…</label>
        - Precedes the form elements and link control text to control
    - <input>
        - type – defines the control type
            - text – single line entry
            - radio – radio buttons
                - Usually for "one of many" selections
                - To get a set of radio buttons to work together, name them all the same
            - checkbox – checkboxes
                - Usually for "multiple option" selections
            - submit – to submit data to defined source (action in <form>)

        - reset – reset fields to default values
        - file – for file attachments
            - Requires additional technology
        - hidden – for transferring data you don’t want the user to readily see
            - Still viewable (in View Source), so do not user for security data
            - Usually used in combination with database-driven sites
        - image – used for image-based buttons
        - password – used for passwords (automatically does asterisks for entry)
        - button – to create generic buttons
    - <select>…</select>
        - For creating drop-down menus
        - Can do single or multiple selections
        - Can show one or more entries at a time
        - <option>…</option> tag used for the menu selections
    - <textarea>…</textarea>
        - Used for multiline text entry
- Other tags
    - <fieldset>…</fieldset> – used for grouping fields for better non-visual rendering
    - <legend>…</legend> – provides caption for <fieldset>
    - <optgroup>…</optgroup> – provides logical group for sets of <option> tags

Note: page 271 of the book is particularly useful when developing forms.

Mixing <form> tags with <table> tags

- Nearly all form elements are usually placed inside of a table for positioning.
- Always place the <form> tag outside the <table> tag.
- Always place the </form> tag outside the </table> tag.
- Notice the location of the form elements in the example below:

```
<html>
<head>
        <title></title>
</head>
<body>

<form name=”form0” method=”post” action=”mailto:rjglotzbach@tech.purdue.edu”>
        <table width=”500” cellpadding=”4” cellspacing=”1” border=”0”>
                    <tr>
                                <td> <input type=”text” name=”FName” size=”25” /> </td>
                    </tr>
                    <tr>
                                <td> <input type=”text” name=”LName” size=”35” /> </td>
                    </tr>
                    <tr>
                                <td> <input type=”submit” value=”Submit”/> </td>
                    </tr>
        </table>
</form>

</body>
</html>
```

- A common mistake that people make is putting the <form> tag inside of the <table> tag.
    - This may cause the form not to send when the submit button is pressed. The form elements would be recognized, but because the <form> tag is improperly nested, the method and action are undefined.
    - Incorrect Example:

```
<table width=”500” cellpadding=”4” cellspacing=”1” border=”0”>
   <form name=”form0” method=”post” action=”mailto:rjglotzbach@tech.purdue.edu”>
```