

HTML

CST 365 – Web Applications
Michael Ruth, Ph.D.
Assistant Professor
Computer Science & I.T.
mruth@roosevelt.edu



Objectives

- Discuss the purpose and use of HTML tables and forms including tags and elements
- Explain the purpose and use of all basic elements and their implementation
- Discuss getting the form to work using the action and method attributes
- Explain the various methods available in terms of purpose and use of each

CST 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189



Michael Ruth, Ph.D.
mruth@roosevelt.edu

2

HTML Topics

- Creating your first HTML Page
- Bold Tag
- Anchor Tag (Links)
- Absolute/Relative File Paths As URLs
- HTML Comments <!-- -->
-
 &
- HTML Tables!
- Cascading Style Sheets!

CST 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189



Michael Ruth, Ph.D.
mruth@roosevelt.edu

3

HTML Tables

- Tables are a means to present columnar information to the user (rows/columns)
 - Tables use the `<table>...</table>` tags
 - Tables are divided into rows (`<tr> ... </tr>`)
 - Each of those rows can be further divided into cells or datums (`<td> ... </td>`)
 - This is how tables create the look of rows/columns

Basic Form Tags

- Everything that goes into a form goes into a top level FORM tag
 - `<form> ... </form>`
- All form elements use the INPUT tag
 - `<input>`
 - The default is a text box
 - **Name** attribute identifies the input received to the receiving application
 - For Example:
 - » `<input name="fname">`
 - » `<input name="lname">`

Form Elements

- There are several form elements which are useful when creating a form
 - Text fields
 - Allows text to be entered
 - Text Areas
 - Allows a lot of text to be entered
 - Check Boxes
 - Allows something to be checked (yes or no)
 - Radio Buttons
 - Allows only one thing (of a group) to be selected
 - Selection Menus
 - A drop-down box which one item is selectable
 - A box which multiple items are selectable
 - Submit and Reset Buttons

Text Fields

- A text field is simply created using the `<input tag>` and TYPE attribute set to "text"
 - Ex: `<input type="text">`
 - Attributes:
 - You can change the default size using the SIZE attribute using the number of characters
 - `<input type="text" size="60">`
 - You can change the max number of acceptable characters using maxlength argument
 - `<input type="text" size="30" maxlength="60">`
 - You can set a default value by using the VALUE attribute
 - `<input type="text" name="fname" value="Michael">`

CSF 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189



Michael Ruth, Ph.D.
mruth@roosevelt.edu
7

Text Areas

- The `<textarea>` tag allows the user to type in a large amount of text at once
 - Ex:
 - `<textarea name="myarea"> </textarea>`
 - The text between the tags specifies the default value of the text area
 - You can set the size of the text area using rows and columns
 - `<textarea rows="2" cols="10"> My name is Michael!</textarea>`

CSF 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189



Michael Ruth, Ph.D.
mruth@roosevelt.edu
8

Check Boxes

- Allows the user to check the box
 - Uses the `<input>` tag as well with the type attribute set to "checkbox"
 - `<input type="checkbox" name="pass">`
 - Another important attribute is checked which is used to denote that the checkbox should be checked
 - `<input type="checkbox" name="pass" checked>`

Do you wish to pass? ☐

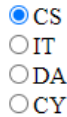
CSF 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189



Michael Ruth, Ph.D.
mruth@roosevelt.edu
9

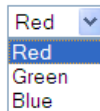
Radio Buttons

- Allows only one thing (of a group) to be selected and also uses the INPUT tag with the type attribute set to "radio"
 - Example:
 - `<input type="radio" name="major" value="CS">CS
`
 - `<input type="radio" name="major" value="IT">IT
`
 - `<input type="radio" name="major" value="DA">DA
`
 - `<input type="radio" name="major" value="CY">CY
`
 - Also can use the checked attribute to denote a default selection



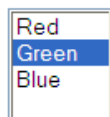
Selection Menus

- Selection menus can be used to give users a set of options to select from
- Uses the **select** tag with a name attribute which surrounds a group of **option** tags
 - Ex:
 - `<select name="color">`
 - `<option value="red">Red</option>`
 - `<option value="green">Green</option>`
 - `<option value="blue">Blue</option>`
 - `</select>`



More Selection Options

- You can "pre-select" a value by setting the selected attribute of the option tag
 - `<option value="green" selected>`
- You can allow the user to select multiple values by setting the multiple attribute of select
 - `<select name="color" multiple>`
 - `<option value="red">Red</option>`
 - `<option value="green">Green</option>`
 - `<option value="blue">Blue</option>`
 - `</select>`



Submit and Reset Buttons

- In order to process the form as a whole, the user must denote that they are finished by clicking the "submit button"
 - This is accomplished using the INPUT tag with the TYPE attribute set to submit
 - The value attribute sets the text which will appear on the button
- The reset button resets the form to its original state and is accomplished using the INPUT tag with the attribute set to reset
 - The value attribute sets the text which will appear on the button
- Ex:
 - `<input type="submit" value="Finished?">`
 - `<input type="reset" value="Start Over?">`



Forms are Required to Work

- There are 2 special attributes which control the behavior of the form after the user hits the submit button
 - METHOD Attribute
 - GET or POST
 - ACTION Attribute
 - This tells the browser where to send the information
 - This will be the script you will write to handle the form
- Ex:
 - `<form method="post" action="myprof.php">`

GET

- The full set of data for the form is appended to the URI specified in the action attribute on the form element
 - All data is followed by a question mark (?)
 - Name/value pairs separated by ampersands (&)
 - Any character in value that would make the URI unworkable (whitespace, etc) is encoded in hexadecimal (ASCII)
 - Name separated from value using equals sign (=)
 - EX:
 - `http://192.168.37.224/ruth/myformscript.php?firstname=michael&lastname=ruth`

Disadvantages of GET

- URIs can get very long
- All fields are sent in plain text
 - Including password fields
 - **Ends in browser history**
- Does not support any character sets other than ASCII
- Form is not entirely necessary!

POST

- The form data is sent in the body of the HTML request to server
 - Still PLAIN TEXT!!!
- Advantages:
 - Can handle other character sets
 - Is not stored/cached in browser memory
 - Uncomplicated URI (simply a normal URI)
- Disadvantage:
 - Out of sight, out of mind!
 - IS NO MORE SECURE THAN GET!

GET or POST?

- Generally, to determine which to use:
 - GET
 - Generally, only used for queries
 - Nothing that will modify data
 - POST
 - In ALL other cases
- Finally, there are times, when even queries should use POST
 - Hide passwords, SSN, etc

Action Attribute

- The action attribute is generally set to a receiver (either a jsp page or a Servlet or a blank page)
 - For now, let's just focus on using a URL

```
<form action="resultpage.html" method="POST">
  <input type="text" name="firstname">
  <input type="submit" value="Submit?">
</form>
```

CS1 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189

 ROOSEVELT
UNIVERSITY

Michael Ruth, Ph.D.
mruth@roosevelt.edu
19

Summary

- Discussed the purpose and use of HTML Forms as well as basic tags and elements
- Explained the purpose and use of all basic elements and their implementation
- Discussed getting the form to work using the action and method attributes
- Explained the various methods available in terms of purpose and use of each

CS1 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189

 ROOSEVELT
UNIVERSITY

Michael Ruth, Ph.D.
mruth@roosevelt.edu
20

Questions?



CS1 365 Web Applications
Advanced HTML (Tables/Forms)
Reading: 189

 ROOSEVELT
UNIVERSITY

Michael Ruth, Ph.D.
mruth@roosevelt.edu
21
