UNIVERSITY OF DIYALA

Web Design and Programming

Lecture 5:

The Forms

Assoc. Prof.
Ali A. Al-Ani

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

XHTML: Forms

- Almost every time we want to collect information from a visitor to our site, we need to use a ***Form***.
- We have probably used several different kinds of forms on different web sites, from simple search boxes, which allow us to enter keywords in order to find what we are looking for, to complex forms that allow us to order groceries or book a holiday online.
- ***XHTML is used only to present the form to the user; it does not allow us to say what happens with that data once it has been collected***.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms: How Forms Work

- ***There are two parts to a working form:***

1. The first part is the form that you see on the page itself that is created using HTML markup. Forms are made up of buttons, input fields, and drop-down menus (collectively known as form controls) used to collect information from the user. Forms may also contain text and other elements.
2. The other component of a web form is an ***application or script*** on the server that processes the information collected by the form and returns an ***appropriate response***. It's what makes the form work.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

<form> Element

- Any form that we create will live inside an element called ***<form>.*** Between the opening ***<form>*** and closing ***</form>*** tags, we will find the ***form controls*** (the ***text input boxes, drop-down boxes, checkboxes, a submit button***, and so on).
- The ***<form>*** element can also contain other markup, like ***paragraphs***, ***headings***, and so on, but a ***<form>*** element ***must not contain*** another ***<form>*** element.
- Our page may contain as many forms as we like. For example, we might have a ***login form***, a ***search form***, and a ***form to subscribe to a newsletter*** all on the same page. If we do have more than one form on a page, users will be able to send the data from only one form at a time to the server.

Department of Computer Science
College of Science

<Form> Element: Attributes

- Every *<form>* element ***should carry at least two attributes***: ***Action and method attributes.*** A *<form>* element may also carry all of the ***universal attributes***, the ***UI event attributes***, and the following attributes:

  Enctype, accept, accept-charset , onsubmit , onreset

1. ***The action Attribute*: *The action attribute indicates what happens to the data when the form is submitted.*** Usually the value of the action attribute is a ***web page or program*** on a web server that will receive the information from this form when a user presses the ***submit button.***

<Form> Element: Attributes

- ***For example:*** if we had a login form consisting of a ***username and password***, the details the user enters may get passed to a page written in ***ASP.net*** on the web server called ***login.aspx***, in which case the action attribute would read as follows:

  <form action="http://www.example.org/membership/login.aspx">

- ***Most browsers will accept only*** a ***URL*** beginning with ***http://*** as the value of the ***action attribute.***

UNIVERSITY OF DIYALA

<Form> Element: Attributes

2. ***The method Attribute*:** Form data can be sent to the ***server*** in two ways, each corresponding to an ***HTTP method***:

1. ***The get method***, which sends data as ***part of the URL***. If the *<form>* element does not carry a method attribute, then by ***default*** the ***get*** method will be used.

- When we send form data to the server using the ***HTTP get method***, the form data is ***appended to the URL*** specified in the action attribute of the ***<form> element***.
- The form data is separated from the URL using a ***question mark***. Following the question mark we get the ***name/value pairs*** for each form control. Each ***name/value*** pair is separated by an ***ampersand (&).***

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

<Form> Element: Attributes

- For example, take the following login form, which we saw when the password form control was introduced:

  http://www.example.com/login.aspx?txtUsername=Bob&pwdPassword=LetMeIn

- The ***great advantages*** of passing form data in a URL is that it can be ***bookmarked***. If we look at searches performed on major search engines such as ***Google***, they tend to use the get method so that the page can be ***bookmarked***. ***The get method, however, has some disadvantages.*** Indeed, when sending ***sensitive data*** such as the password shown here, or credit card details, we ***should not use the get method*** because the sensitive data becomes ***part of the URL*** and is in full view to everyone (and could be bookmarked).

Department of Computer Science
College of Science

<Form> Element: Attributes

- ***We should not use the HTTP get method when:***

1. ***We are updating a data source*** such as a database or spreadsheet (because someone could make up URLs that would alter our data source).
2. ***We are dealing with sensitive information***, such as passwords or credit card details.
3. ***We have large amounts of data*** (because older browsers do not allow URLs to exceed more than 1,024 characters).
4. ***The form contains a file upload control*** (uploaded files cannot be passed in the URL).
5. ***Our users might enter non-ASCII characters*** such as ***Hebrew* or *Cyrillic*** characters.

<Form> Element: Attributes

2. ***The post method:*** In this method the form ***data will hides in the HTTP headers***, When we send data from a form to the server using the HTTP post method, the form data is sent ***transparently in what is known as the HTTP headers***.

- If the login form we just saw was sent using the post method, it could look something like this in the HTTP headers:

```
User-agent: MSIE 5.5
Content-Type: application/x-www-form-urlencoded
Content-length: 35
...other headers go here...
txtUserName=Bob&pwdPassword=LetMeIn
```

UNIVERSITY OF DIYALA

<Form> Element: Attributes

3. ***The id attribute*** allows us to ***unique identify the <form> element*** within a page, just as we can use it to uniquely identify any element on a page.

- It is good practice to ***give every <form> element*** *an* ***id attribute***, because many forms make use of style sheets and scripts, which may require the use of the ***id attribute*** to identify the ***form***.
- The value of the id attribute should be ***unique*** within the document, for example, ***frmLogin or frmSearch***.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

<Form> Element: Attributes

4. ***The Onsubmit attribute:*** When we have probably filled in a form on a web site, and then, as soon as we have clicked the button to send the form data, been shown a message telling us that ***we have missed entering some data, or entered the wrong data***.

- When this happens, the chances are we have come across a form that uses the ***onsubmit*** attribute to ***run a script*** in the ***browser*** that checks the data we entered ***before the form is sent to the server***. When a user clicks a submit button, something called an ***event fires***.
- The idea behind these events is that a script (such as a ***JavaScript script***) can be run before the data is sent to the server ***to ensure the quality and accuracy of the submitted data.***

Department of Computer Science
College of Science

<Form> Element: Attributes

- The value of the ***onsubmit*** attribute should be a script function that would be used when this event fires. So, an ***onsubmit attribute*** on the ***<form> element*** might look like this:

  onsubmit="validateFormDetails();"

- In this case, the ***validateFormDetails()*** function should have been defined in the document already (probably in the ***<head> element***). So when the user clicks the submit button, this ***function will be called and run.***

<Form> Element: Attribute

- ***There are two advantages to making some checks on the form before sent to the server:***
  1. ***The user does not have to wait the extra time it would take for the page to be sent to the server and then returned if there are any errors.***
  2. ***The server does not have to deal with as much error checking as it would if the checks by the browser had not been performed.***

5. ***The onreset*:** Some forms contain a reset button that empties the form, although the button might say something like clear form instead; when this button is pressed, an ***onreset*** event fires and a ***script can be run***. When the ***onreset*** attribute is used, its value is a script (as with the ***onsubmit*** attribute) that is executed when the user clicks the button that calls it

UNIVERSITY OF DIYALA

Forms Controls

- This section covers the different types of ***form controls*** that we can use to collect data from a visitor to our site. We will see:

1. ***Text input controls***
2. ***Buttons***
3. ***Checkboxes and radio buttons***
4. ***Select boxes (sometimes referred to as drop-down menus) and list boxes***
5. ***File select boxes***

UNIVERSITY OF DIYALA

Forms Controls: Text input controls

- Possibly the most famous text input box is the one right in the middle of the Google home page. There are actually three types of text input used on forms:

1. ***Single-line text input controls:*** Used for items that require only one line of user input, such as ***search boxes*** or e-mail addresses. They are created using the ***<input> element***.
2. ***Password input controls*:** These are just like the single-line text input, except they ***mask the characters***. They are also created using the ***<input> element***.
3. ***Multi-line text input controls:*** Used when the user is required to give details that may be ***longer than a single sentence***. Multi-line input controls are created with the ***<textarea> element***.

UNIVERSITY OF DIYALA

Forms Controls: Text input controls

1. ***Single-line text input controls:*** are created using an ***<input> element*** *whose* ***type attribute*** has a value of ***text***.

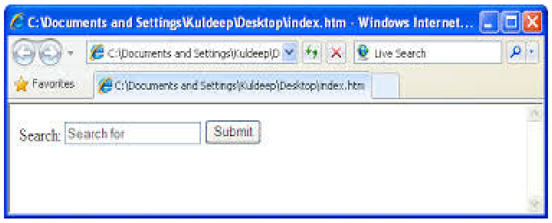
<form action="http://www.example.com/search.aspx" method="get" name="frmSearch">

<p> Search: /<p>

<input type="text" name="txtSearch" value="Search for" size="20" maxlength="64" />

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

</form>


C:\Documents and Settings\Kuldeep\Desktop\index.htm - Windows Internet...
Live Search
Favorites
C:\Documents and Settings\Kuldeep\Desktop\index.htm
Search:
Search for
Submit

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms Controls: Text input controls

2. ***Password input controls:*** If we want to collect sensitive data such as passwords and credit card information, we should use the ***password input***. For example:

<form action="http://www.example.com/login.aspx" method="post">

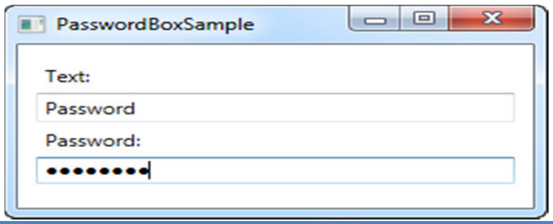
<p>Username:</p>

<input type="text" name="txtUsername" value="" size="20" maxlength="20" />

<br /> <p> Password: </p>

<input type="password" name="pwdPassword" />

</form>



Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms Controls: Multiple-Line Text Input Controls

3. ***Multi-line text input controls:*** If we want to allow a visitor to our site to enter more than one line of text, we should create a multiple line text input control using the ***<textarea> element.*** For example:

<form action="http://www.example.org/feedback.asp" method="post">

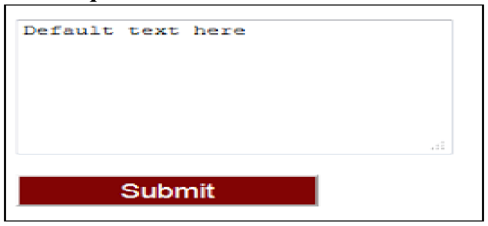
<p>Please tell us what you think of the site and then click submit: </p> <br />

<textarea name="txtFeedback" rows="20" cols="50">

Enter your feedback here. </textarea>

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

</form>



Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms Controls: Buttons

- Buttons are most commonly used to ***submit a form***, although they are sometimes used to ***clear or reset*** a form and even to trigger client-side scripts. We can create a button in three ways:

1. Using an ***<input> element*** with a ***type attribute*** whose ***value*** is ***submit***, ***reset***, or ***button***
2. Using an ***<input> element*** with a ***type*** attribute whose ***value*** is ***image***
3. Using a ***<button> element***

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms Controls: Buttons

- When we use the ***<input> element*** to create a button, the type of button we create is specified using the ***type attribute***. The type attribute can take the following values:

1. ***Submit***: which creates a button that ***automatically submits a form.***
2. ***Reset***: which creates a button that ***automatically resets form controls to their initial values.***
3. ***Button***: which creates a button that ***is used to trigger a client-side script*** when the user clicks that button. ***onclick attribute*** **u**sed to trigger a script when the user clicks the button; the value of this attribute is the script that should be run. we can also trigger a script when the button gains or loses focus with the ***onfocus*** and ***onblur*** event attributes.

UNIVERSITY OF DIYALA

Forms Controls: Checkboxes

- ***Checkboxes*** are just like the little boxes that we have to check on paper forms. ***Checkboxes*** can appear individually, with each having its own name, or they can appear as a group of checkboxes that share a ***control name*** and allow users to select several values for the same property. Checkboxes are ideal form controls when we need to allow a user to:

1. ***Provide a simple yes or no response with one control (such as accepting terms and conditions or subscribing to an e-mail list)***
2. ***Select several items from a list of possible options (such as when you want a user to indicate all of the skills they have from a given list)***

UNIVERSITY OF DIYALA

Forms Controls: Checkboxes

- A checkbox is created using the ***<input> element*** whose type attribute has a value of ***checkbox***. Following is an example of some checkboxes that use the same control name:

```
<form action=”http://www.example.com/cv.aspx” method=”get” name=”frmCV”>
<p>Which of the following skills do you possess? Select all that apply.</p>
<input type=”checkbox” name=”chkSkills” value=”html” checked=”checked” />
<p>HTML </p> <br />
<input type=”checkbox” name=”chkSkills” value=”CSS” />CSS<br />
</form>
```

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms Controls: Radio Buttons

- ***Radio buttons*** are similar to checkboxes in that they can be either ***on or off***, but there are two key differences:

1. ***When we have a group of radio buttons that share the same name, only one of them can be selected.***
2. ***We should not use radio buttons for a single form control where the control indicates on or off because once a lone radio button has been selected it cannot be deselected again (without writing a script to do that).***

- Therefore, ***radio buttons are ideal if we want to provide users with a number of options from which they can pick only one***.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms Controls: Radio Buttons

- The ***<input> element*** is again called upon to create radio buttons, and this time the type attribute should be given a value of radio. For example:

```
<form action=”http://www.example.com/flights.aspx” name=”frmFlight” method=”get”>
<p>Please select which class of travel you wish to fly: </p><br />
<input type=”radio” name=”radClass” value=”First” /> <p>First class </p><br />
<input type=”radio” name=”radClass” value=”Business” /> <p> Business class </p> <br />
</form>
```

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms: Select Boxes

- ***A drop-down select box*** allows users to ***select one item from a drop-down menu***. Drop-down select boxes can take ***up far less space*** than a group of radio buttons.
- Drop-down select boxes can also provide an alternative to single-line text input controls where we want to limit the options that a user can enter.
- ***For example, we can use a select box to allow users to indicate which country or state they live in.***
- A drop-down select box is contained by a ***<select> element***, while each individual option within that list is contained within an ***<option> element***.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms: Select Boxes

- For example, the following form creates a drop-down select box:

<select name=”selColor”>

<option selected=”selected” value=””>Select color</option>

<option value=”red”>Red</option>

<option value=”green”>Green</option>

<option value=”blue”>Blue</option>

</select>

- The text between the ***opening <option> element*** and the ***closing </option> tags*** is used to display options to the user, while the value that would be sent to the server if that option is selected is given in the value attribute.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

Forms: File Select Boxes

- If we want to allow a user to ***upload a file*** to our web site from his or her computer, we will need to use a file upload box, also known as a ***file select box***. This is created using the ***<input> element*** , but this time we give *the* ***type attribute*** a value of file

<form action=”http://www.example.com/imageUpload.aspx” method=”post”

name=”fromImageUpload” enctype=”multipart/form-data”>

<input type=”file” name=”fileUpload” accept=”image/*” />

<br /><br /><input type=”submit” value=”Submit” />

</form>

- When we are using a file upload box, the ***method attribute*** of the ***<form> element*** must be ***post***.

Department of Computer Science
College of Science

UNIVERSITY OF DIYALA

The End

Department of Computer Science
College of Science