CITS1231 Web Technologies

HTML Forms
Introducing Web Forms

- **Web forms** collect information from customers
- Web forms include different **control elements** including:
  - Input boxes
  - Selection lists
  - Drop-down lists boxes
  - Option buttons or radio buttons
  - Check boxes
  - Group boxes
  - Text areas
  - Form buttons
Forms

• Forms are often used in HTML to collect and process data using client side and server-side software.

• While HTML supports the creation of forms, it does not include tools to process the information.

• A form specified in HTML has no functionality without an accompanying program. The information can be processed through a program running on a Web server. Server-based programs are written in many languages
  
  – Common Gateway Interface (CGI) scripts that are written in perl, PHP, TCL, C/C++, Visual Basic

• We will look at the elements used in creating forms and will cover the client-side programming in Javascript lectures.
A Sample Form

Text field accepting plain text

Your Name: <input type="text" name="yourName" value="Your Name">

Multiple Line Text Field

Your comments go here.<textarea name="comments" rows=3 cols=40"></textarea>

Nonexclusive Checkbox Group

Your favourite things: <input type="checkbox" name="things" value="Book"> Book
<input type="checkbox" name="things" value="Game"> Game
<input type="checkbox" name="things" value="Music"> Music

Exclusive Radio Buttons Group

Your profession: <input type="radio" name="profession" value="Student"> Student
<input type="radio" name="profession" value="Teacher"> Teacher
<input type="radio" name="profession" value="Doctor"> Doctor
<input type="radio" name="profession" value="Not telling"> Not telling

Single Selection

Your favourite colour: <select name="favouriteColour">
  <option value="Red">Red</option>
  <option value="Green">Green</option>
  <option value="Blue">Blue</option>
</select>

Multiple Selection

Your favourite food for lunch: <input type="checkbox" name="lunchOptions" value="Biscuits"> Biscuits
<input type="checkbox" name="lunchOptions" value="Sandwich"> Sandwich
<input type="checkbox" name="lunchOptions" value="Noodles"> Noodles

Send  Cancel
A Sample Form

Text field accepting plain text
Your Name:

Multiple Line Text Field
Your comments go here.

Nonexclusive Checkbox Group
Your favourite things:  □ Books  □ Games  □ Music

Exclusive Radio Buttons Group
Your profession:  □ Student  □ Teacher  □ Doctor  □ Not telling

Single Selection
Your favourite colour:  

Multiple Selection
Your favourite food for lunch:  

The Form Element

- Forms are created using the form element, structured as follows:

  `<form attributes>
   elements
  </form>`

where

- attributes are the attributes that control how the form is processed and
- elements are elements places within the form.
Form Attributes – Name and ID

- Always specify an id or name for the form
- Two attributes are available to identify the form: id and name
  
  
  ```html
  <form name="name" id="id">... </form>
  ```

  where name is the name of the form and id is the id of the form.
Form Attributes – method and action

• Form data is sent as name/value pairs:

  name1=value1 & name2=value2 & ...

• The form element has two attributes
  
  - method – indicates how the data collected by the form should be transmitted to the server using the HTTP
    
    • GET – the form data is appended to the URL
    
    • POST – the form data is sent as a separate message
  
  - action – what to do when the input type “submit” is pressed (more on this when we talk about programming)
Forms: GET or POST?

• Use GET if:
  – Interaction is more like a question
  – It is a safe operation such as a query, read operation, or lookup

• Use POST if:
  – Interaction is more like an order, or
  – Interaction changes state of the resource (e.g., payment transaction), or
  – The user will be held accountable for the interaction.
Form Method: GET

- Form data appended to URL.
- URL is followed by “?” and then the name/value pairs.


- GET requests can be cached
- GET requests can remain in browser history
- GET requests can be bookmarked
- GET requests can be distributed & shared
- GET requests can be hacked
  - Don’t send password or sensitive data with GET.
Form Method: POST

• Form data sent as a separate message

• Use for sensitive data such as
  – passwords
  – credit card numbers
  – bank account numbers

• IE has max url length of 2048 characters. So if lots of data, use POST instead of GET even if not sensitive.
Form Method: Example

• In the following example, the form data is sent as POST request to a program called server_side_processing.php in the same directory as the current form on the server.

  <form method="POST" action="server_side_processing.php">

• We will explore server side PHP scripts later in the semester.
Using the mailto Action

- The mailto action accesses the user’s own e-mail program and uses it to mail form information to a specified e-mail address
  - By-passes the need for server-based programs
- The syntax is as follows:
  
  ```html
  <form action="mailto:e-mail_address" method="post"
  enctype="text/plain"> ... </form>
  
  where
  - `e-mail_address` is the e-mail address of the recipient in the form
Creating Input Boxes

To create an input box, use the following HTML code:

```html
<input type="type" name="name" id="id" value="value" size="value" maxlength="value" />
```

where

- type specifies the type of input field
- the name and id attributes identify the field,
- the value attribute assigns the field’s default value,
- the size attribute defines the width of the input box in characters (default 20 characters), and
- the maxlength attribute specifies the maximum number of characters that a user can enter into the field.
Types of an input box

- The sample form HTML code illustrates several input elements with several different type attributes
  - **text** – Single line data entry textbox
  - **checkbox** – a simple tick/check box
  - **checkbox** – with a group of items with the same name but different values, indicate **non-exclusive** multiple selections from a list of items
  - **radio** – with a group of items with the same name but different values, indicate **exclusive** selection from a list of items.
  - **submit** – a button to submit the form for the action file to process
  - **reset** – a button to reset the form to defaults.
Creating Input Boxes

- Input types:
  - type="image"
    - Displays an input image that can be clicked to perform an action from a script
  - type="password"
    - Displays an input box that hides text entered by the user
  - type="button"
    - Displays a button that can be clicked to perform an action from a script
  - type="file"
    - Displays a browse button to locate and select a file
  - type="hidden"
    - Creates a hidden field, not viewable on the form
Creating Check Box

• To create a check box, use:

```html
<input type="checkbox" name="name" id="id"
    value="value" />
```

where

– the name and id attributes identify the check box field and
– the value attribute specifies the value sent to the server if the check box is selected

• To specify that a check box be selected by default, use the checked attribute as follows:

```html
<input type="checkbox" checked="checked" />
```

or

```html
<input type="checkbox" checked />
```
Creating Radio buttons

- Option buttons, or radio buttons allow users to make selections.

- Unlike checkboxes, selection lists, radio buttons only allow the user to select one option at a time.

All radio buttons belonging to the same field share a common name.

```html
<input type="radio" name="party" id="dem" value="dem" />
<label for="dem">Democrat</label>
<br />
<input type="radio" name="party" id="gop" value="gop" />
<label for="gop">Republican</label>
<br />
<input type="radio" name="party" id="ind" value="ind" />
<label for="ind">Independent</label>
```

Labels are matched to the id values of the option buttons.
Working with Form Labels

• You can also expressly link a label with an associated text element for scripting purposes

• The syntax for creating a form label is as follows:

  <label for="id">label text</label>

where

  – id is the value of the id attribute for a field on the form, and
  – label text is the text of the label.
Elements of a form (cont.)

• Two other ways of collecting data are also illustrated in our example form:
  - **textarea** – a multi-line textbox allows multiple lines input, number of rows and columns can be specified.
  - **select**
    • Without **multiple** – produces a **combo-box** for selecting a single item
    • With **multiple** – produces a **list-box** for selecting multiple items (in Windows, holding the CTRL key down while selecting more than one items.

• There are other ways to allow user entering data using forms, For example, **button, image** and **file upload**.
Creating a text area box

To create a text area box, use the textarea element:

```html
<textarea name="name" id="id" rows="value" cols="value">
  default text
</textarea>
```

where

- the rows and cols attributes define the dimensions of the input box and
- the rows attribute indicates the number of lines in the input box
Creating a Text Area Box

<label for="comments">Comments?</label> <br />
<textarea name="comments" id="comments" rows="5" cols="50">
Enter comments here.
</textarea>
Creating a Selection List

- **A selection list** is a list box from which a user selects a particular value or set of values
  - Selection lists are useful when there are a fixed set of possible responses from the user

- You can create a selection list using the `<select>` tag

- You can specify each individual selection item using the `<option>` tag

- You can change the number of options displayed in the selection list by modifying the size attribute. The syntax is as follows:
  - `<select size="value">... </select>`

  where value is the number of items that the selection list displays in the form.
Modifying Appearance of a Selection List

- size = “1”
- size = “4”
- size = “7”
- size = “9”
Working with Option Groups

- Use **option groups** to organize selection lists into distinct groups.

```html
<select attributes>
<optgroup label="label1">
  <option>itema1</option>
  <option>itema2</option>
...
</optgroup>
...
</select>
```
Option Groups: Example

```html
<html>
<body>
<select>
  <optgroup label="Fruit">
    <option value="apple">Apple</option>
    <option value="pear">Pear</option>
  </optgroup>
  <optgroup label="Vegetable">
    <option value="carrot">Carrot</option>
    <option value="turnip">Turnip</option>
  </optgroup>
</select>
</body>
</html>
```
Creating a Field Set

- HTML and XHML allow you to organize radio buttons into a group of fields called **field sets**
  - Most browsers place a **group box** around a field set to indicate that the fields belong to a common group
    
    `<fieldset>
    fields
    </fieldset>

    where *fields* are the individual fields within a set.

- To add a caption to a field set, add the following tag after the opening `<fieldset>` tag:
  
  `<legend>text</legend>`

  where *text* is the text of the field set caption.
Field Set: Example

```html
<html>
<body>
<form>
<fieldset>
  <legend>Contact 1:</legend>
  Name: <input type="text" size="30" />
  Email: <input type="text" size="30" />
</fieldset>
<fieldset>
  <legend>Contact 2:</legend>
  Name: <input type="text" size="30" />
  Email: <input type="text" size="30" />
</fieldset>
</form>
</body>
</html>
```
Form Buttons

- Buttons are a type of control element that performs an action
- Types of buttons:
  - Command button
  - Submit button
  - Reset button
  - File button
Creating a Command button

- **Command buttons** are created using the `<input>` tag:
  
  ```html
  <input type="button" value="text" />
  ```

- **Submit buttons** submit forms to the server for processing when clicked. Syntax is as follows:
  
  ```html
  <input type="submit" value="text" />
  ```

- **Reset buttons** reset forms to their original (default) values. Syntax is as follows:
  
  ```html
  <input type="reset" value="text" />
  ```
Completed Registration Form
Designing a Command button

- Use the button element for greater artistic control over the appearance of a button

  `<button name="name" id="id" value="value" type="type">`  
  `content`  
  `</button>`

  - where the name and value attributes specify the name of the button and the value sent to a server-based program,  
  - the id attribute specifies the button’s id,  
  - the type attribute specifies the button type, and  
  - the content is page content displayed within the button.
Creating a File button

- **File buttons** are used to select files so that their contents can be submitted for processing to a program.

- The Web page then only displays the file’s location, not the file’s contents.
Tips for Creating Effective Forms

• Label all control elements clearly and concisely
• Use horizontal lines, tables, and line breaks to separate topical groups from one another
• Use field sets to organize common groups of fields, especially radio buttons
• Use tabindex attribute to ensure that users will move correctly from one field to another (tabindex=“value”)
Tips for Creating Effective Forms

• Use radio buttons, check boxes, and selection lists whenever possible to limit a user’s choice of entries, thus reducing the chance of an erroneous data value. Use input boxes only when the field has no predefined list of values.

• Use selection lists for items with several possible options. Use radio buttons for items with few options. Use a check box for each item with only two possible values.
Tips for Creating Effective Forms

• Let users know the correct format for input box text by inserting default text in the appropriate format (for example, insert the text string, “mm/dd/yyyy” in a Date input box to indicate the format for inserting date values

• Use password fields for sensitive or confidential information (such as passwords)

• Because form elements differ between browsers, view your form on different browsers and different browser versions to ensure that the form displays correctly in all situations

• Useful site for more information:

  http://www.456bereastreet.com/archive/200701/styling_form_controls_with_css_revisited/