Web Design
Course

LECTURE EIGHT
Introduction to HTML and XHTML

**Computer Science
Collage of Education
AL_Mustansyria University
Forth Year**

2018-2019

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- checkbox.html
     An example to illustrate a checkbox
     -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Checkboxes </title>
  </head>
  <body>
    <p>
      Grocery Checklist
    </p>
    <form action = "">
      <p>
        <label> <input type = "checkbox"  name = "groceries"
              value = "milk"  checked = "checked" /> Milk </label>
        <label> <input type = "checkbox"  name = "groceries"
              value = "bread" /> Bread </label>
        <label> <input type = "checkbox"  name = "groceries"
              value = "eggs" /> Eggs </label>
      </p>
    </form>
  </body>
</html>
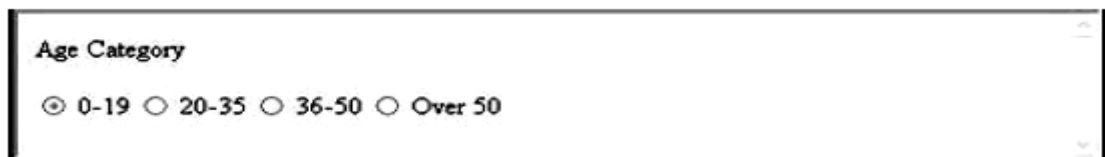```

Grocery Checklist

☑ Milk ☐ Bread ☐ Eggs

Figure 22 Display of checkbox.html

**If the user does not turn on any of the checkbox buttons in our example, milk will be the value for groceries in the form data. If the milk checkbox is left on and the eggs checkbox is also turned on by the user, the values of groceries in the form data would be milk and eggs. Radio buttons are closely related to checkbox buttons. The difference between a group of radio buttons and a group of checkboxes is that only one radio button can be on or pressed at any time. Every time a radio button is pressed, the button in the group that was previously on is turned off. Radio buttons are named after the mechanical push buttons on the radios of cars of the 1950s—when you pushed one button on such a radio, the previously pushed button was mechanically forced out. The type value for radio buttons is radio. All radio buttons in a group must have the name attribute set in the <input> tag, and all radio buttons in a group must have the same name value. A radio button definition may specify which button is to be initially in the pressed, or on, state. This specification is indicated by including the checked attribute, set to the value checked, in the <input> tag of the button's definition. If no radio button in a group is specified as being checked, the browser usually checks the first button in the group. The following example illustrates radio buttons:**

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- radio.html
      An example to illustrate radio buttons
      -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Radio </title>
  </head>
  <body>
    <p>
      Age Category
    </p>
    <form action = "">
      <p>
        <label><input type = "radio" name = "age"
               value = "under20" checked = "checked" />
               0-19 </label>
        <label><input type = "radio" name = "age"
               value = "20-35" /> 20-35 </label>
        <label><input type = "radio" name = "age"
               value = "36-50" /> 36-50 </label>
        <label><input type = "radio" name = "age"
               value = "over50" /> Over 50 </label>
      </p>
    </form>
  </body>
</html>
```
Figure 23 shows a browser display of radio.html.



**Figure 23 Display of radio.html**

## 7.3 The <select> Tag

**Checkboxes and radio buttons are effective methods for collecting multiple-choice data from a user. However, if the number of choices is large, the form becomes too long to display. In these cases, a menu should be used. A menu is specified with a <select> tag (rather than with the <input> tag). There are two kinds of menus: those in which only one menu item can be selected at a time (which are related to radio buttons) and those in which multiple menu items can be selected at a time (which are related to checkboxes). The default option is the one related to radio buttons. The other option can be specified by adding the multiple attribute, set to the value "multiple". When only one menu item is selected, the value sent in the form data is the value of the name attribute of the <select> tag and the chosen menu item. When multiple menu items are selected, the value for the menu in the form data includes all selected menu items. If no menu item is selected, no value for the menu is included in the form data. The name attribute, of course, is required in the <select> tag. The size attribute, specifying the number of menu items that are to be displayed for the user, can be included in the <select> tag. If no size attribute is specified, the value 1 is used. If the value for the size attribute is 1 and multiple is not specified, just one menu item is displayed, with a downward scroll arrow. If the scroll arrow is clicked, the menu is displayed as a pop-up menu. If either multiple is specified or the size attribute is set to a number larger than 1, the menu is usually displayed as a scrolled list.**

**Each of the items in a menu is specified with an <option> tag, nested in the select element. The content of an <option> tag is the value of the menu item, which is just text. (No tags may be included.) The <option> tag can include the selected attribute, which specifies that the item is preselected. The value assigned to select is "selected", which can be overridden by the user. The following document describes a menu with the default value (1) for size:**

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- menu.html
        An example to illustrate menus
        -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Menu </title>
  </head>
  <body>
    <p>
      Grocery Menu - milk, bread, eggs, cheese
    </p>
    <form action = "">
      <p>
        With size = 1 (the default)
        <select name = "groceries">
          <option> milk </option>
          <option> bread </option>
          <option> eggs </option>
          <option> cheese </option>
        </select>
      </p>
    </form>
  </body>
</html>
```

Figure 24 shows a browser display of menu.html. Figure 25 shows a browser display of menu.html after clicking the scroll arrow. Figure 26 shows a browser display of menu.html after modification to set size to "2."
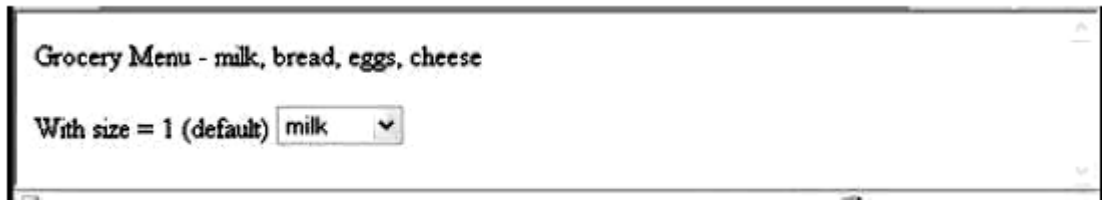
Figure 24 Display of menu.html (default size of 1)

Figure 25 Display of menu.html after the scroll arrow is clicked

Figure 26 Display of menu.html with size set to 2

When the multiple attribute of the <select> tag is set, adjacent options can be chosen by dragging the mouse cursor over them while the left mouse button is held down. Nonadjacent options can be selected by clicking them while holding down the keyboard Control key.

7.4 The <textarea> Tag

In some situations, a multiline text area is needed. The <textarea> tag is used to create such a control. The text typed into the area created by <textarea> is not limited in length, and there is implicit scrolling when needed, both vertically and horizontally. The default size of the visible part of the text in a text area is often quite small, so the rows and cols attributes should usually be included and set to reasonable sizes. If some default text is to be included in the text area, it can be included as the content of the text area element. The following document describes a text area whose window is 40 columns wide and three lines tall:

_____

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- textarea.html
     An example to illustrate a textarea
     -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Textarea </title>
  </head>
  <body>
    <p>
      Please provide your employment aspirations
    </p>
    <form action = "handler">
      <p>
        <textarea name = "aspirations"  rows = "3"  cols = "40">
          (Be brief and concise)
        </textarea>
      </p>
    </form>
  </body>
</html>
```

Please provide your employment aspirations

(Be brief and concise)

Figure 27 Display of textarea.html after some text entry

7.5 The Action Buttons

The Reset button clears all of the controls in the form to their initial states. The Submit button has two actions: First, the form data is encoded and sent to the server; second, the server is requested to execute the server-resident program specified in the action attribute of the <form> tag. The purpose of such a server-resident program is to process the form data and return some response to the user. Every form requires a Submit button. The Submit and Reset buttons are created with the <input> tag, as shown in the following example:

```
<form action = "">
<p>
<input type = "submit" value = "Submit Form" />
<input type = "reset" value = "Reset Form" />
</p>
</form>
```

Submit Form   Reset Form

A plain button has the type button. Plain buttons are used to choose an action.