HTML Forms
OBJECTIVES

In this unit, you will learn how to:

• Create a form in an HTML page

• Create input fields within a form such as text boxes and menus
HTML forms

• HTML forms allow the user to enter information to be sent to the web server

• A form can contain ordinary text and images, but also interactive elements for input
  – Text boxes
  – Menus
  – Checkboxes and “radio buttons” (single choice)
  – Submit and Cancel buttons
Form submission

• When form is submitted, data is sent to server
• Typically processed by CGI program or dynamic server code such as PHP
• Information represented by name-value pairs
  – name = name of form field
  – value = value provided or selected by user
Defining a form

• A form is defined by:

```html
<form name="form-name" action="handler-URL"
method="submit-method">
... (form area)
</form>
```

• Any normal HTML elements, as well as form input fields, can go in form area.
Form name

- Form name attribute is optional, but providing it allows JavaScript to access the form fields
Form **action**

- Form **action** attribute is URL of handler on server that will receive the data
  - May be omitted or "#", implying URL is same as document itself. (Handler needs code to distinguish between initial and response situations.)
Form method

- Form **method** attribute is either "get" or "post"
  - Default is "get". Data sent as appendage to action URL. Server passes data to handler via an environment variable. Use only for very simple situations.
  - Generally use "post". Data sent in a data body along with request to action URL. Server passes data to handler via an input method.
Form **input** field

- Used to define text boxes, checkboxes, etc.
  
  ```html
  <input name="field-name" type="input-type" ... />
  ```

  - **name** identifies the data item when sent to server.
  - Also use **name** to access the element from JavaScript.
  - **type** specifies what kind of input element. Different types may take additional attributes.
input field type

- **type** may be:
  - "text" – small box where user can type text
  - "password" – like "text" but typed text is hidden
  - "checkbox" – clickable box to check or uncheck
  - "radio" – like "checkbox" but used in group of fields with same **name**, only one of which may be selected
Special input types

- Other values of type attribute:
  - "submit" – clicking it causes form to be submitted
  - "reset" – restores form fields to initial values
  - "button" – provides a button to which JavaScript actions can be tied
  - "hidden" – field does not appear in rendered page.
    - This was used to carry data along for continuity in a visit
    - Need is better handled other ways nowadays
New HTML5 input types

- HTML5 added several flavors of text box for specific kinds of information
  - Contents automatically validated when form is submitted.
  - Examples: "date", "tel", "email"
  - Support for these in browsers is currently spotty. We will see how to validate field values ourselves.
Other **input** attributes

- **size**="*width*" for input fields of type **text** or **password**. The *width* is an integer giving the width of the box in characters. Default width is 20.

- **maxlength**="*length*" for fields of type **text** or **password**. Specifies the maximum number of characters that can be entered into the box.

- **checked** for **checkbox** and **radio** buttons. (Not necessary to assign a value. XHTML required value **checked**="*checked*".) If this attribute is present, it means the choice is selected by default.

- **value**="*default-value*" specifies a default value for the item or, for **submit** and **reset** buttons, a label for the button.
Form menus

• A menu is defined by:

\[
\text{<select name="menu-name">}
\text{  <option>First choice</option>}
\text{  <option>Second choice</option>}
\text{  ...}
\text{</select>}
\]
Form **select** menu attributes

- By default, a **select** menu appears as a window with a single item appearing, and a scroll button to allow selection of desired item.
- Use `size="length"` to increase the number of items showing in the window.
- Add attribute **multiple** to allow more than one item to be selected at a time.
- Add attribute **selected** to an **option** to make it selected by default.
The **textarea** field

- The **input** textbox is meant for only short pieces of text. For larger amounts (e.g. user comments) use
  
  `<textarea name="textarea-name" rows="rows" cols="cols"> 
  Default text can go here. 
  </textarea>`

- The **rows** and **cols** specify height and width of text box.

**Example**
Form `label` element

- The `label` element typically encloses labeling text for a form input field.
  - No difference in rendering.
  - Extends mouse functionality for clickable inputs to include the label text.

- Association of label with field may be
  - *Implicit*: by enclosing the input field inside the label along with the label text.
  - *Explicit*: using `for="id"` attribute to associate with field having that `id`. Label may be located anywhere in form.

Example
The **id** attribute

- The last example introduced the **id** attribute. It is used to uniquely identify any element of a document.

  **Example:**

  ```html
  <input name="foo" id="foo" type="text" />
  ```

  - The difference in brief: **name** is for use in form data submitted to server, **id** is for use as DOM element (in document internal cross-references or in JavaScript). It is usual to supply both.
  - **id** must be unique within document, **name** need only be unique within form.
  - Except for radio buttons, best to use same **name** as **id** for form fields. (Radio buttons in a group share same **name** but must have different **ids**.)
Form **fieldset** element

- Fields in a form may be grouped together inside a fieldset element.
- Browser draws a box around the fieldset.
- May be useful for JavaScript to access elements of fieldset.

Example