Introduction to Web Design
Lecture_3

Dr. Sarah mohamed

Forms in HTML

- HTML forms provide a great way to capture data from your web site. A form in an HTML page can be used to submit data to a server for further processing.
- Each HTML form is introduced by the tag and is made up of different input elements. The tag has several attributes that would tell the browser what to do with the information that is entered in the different fields. These attributes are:

  Action (The action attribute tells your browser the location of the script that you are going to use to process the form. Or Specifies where to send the form-data when a form is submitted)

  method (the method attribute has two possible values, method=get or method=post, both of which specify a method of submitting data to the script you have specified in the action attribute)

<Form> element syntax

```
<form action = "……." target= "……..">
        ……….
        form elements
        ……….
</form>
```

```
<form action="/action_page.php">

</form>
```

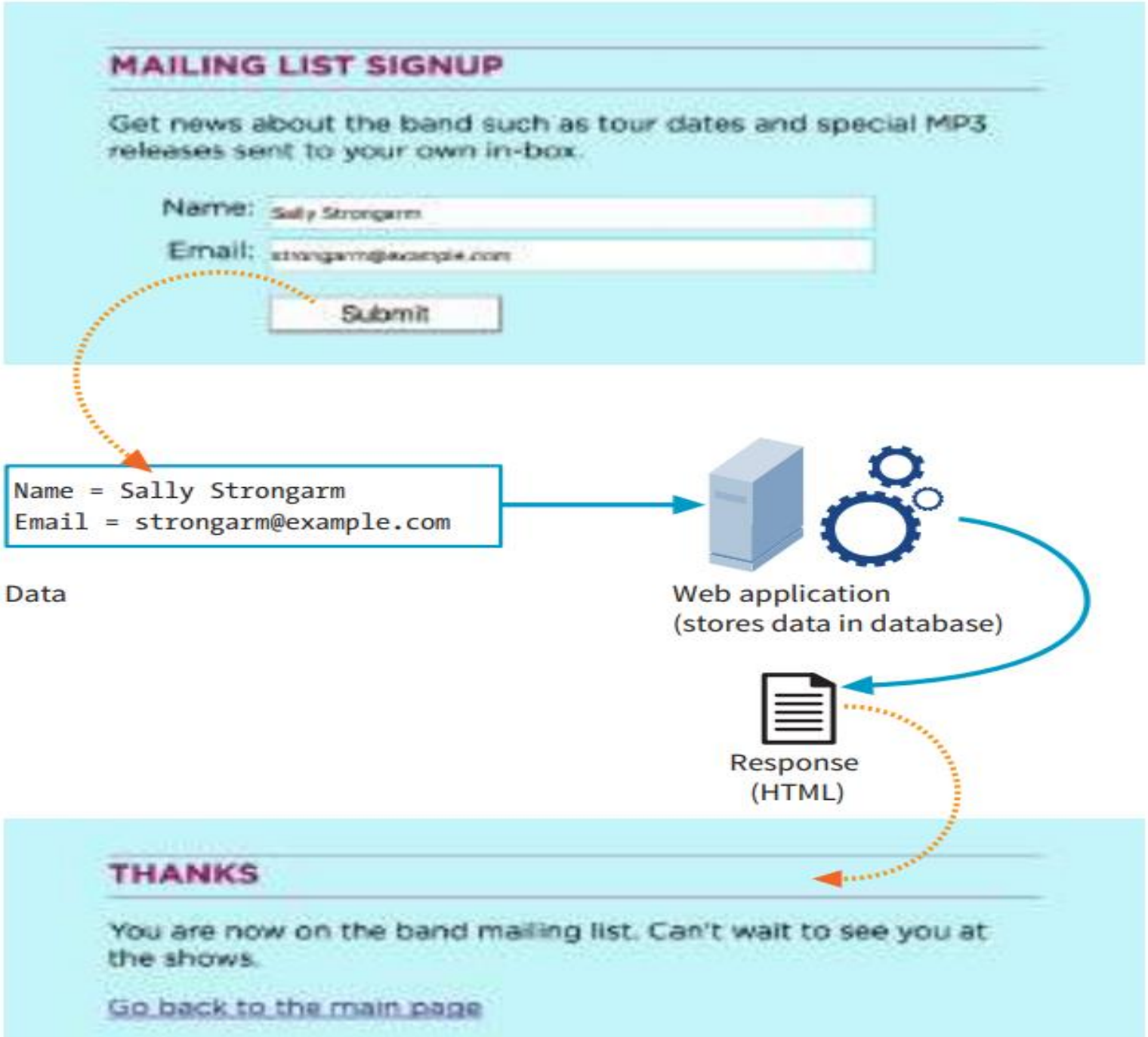
The method attribute specifies the HTTP method to be used when submitting the form data .

- GET: The default method when submitting form data, the submitted form data will be visible in the page address field
- POST: does not display the submitted form data in the page address field, used if the form data contains sensitive or personal information.

There are also other input methods that may be used in an HTML form, including dropdown lists, radio buttons, check boxes, text areas and others. Let us see how each of these text elements is used in an HTML form

- Text fields
- Checkboxes,
- Radio buttons,
- Submit buttons

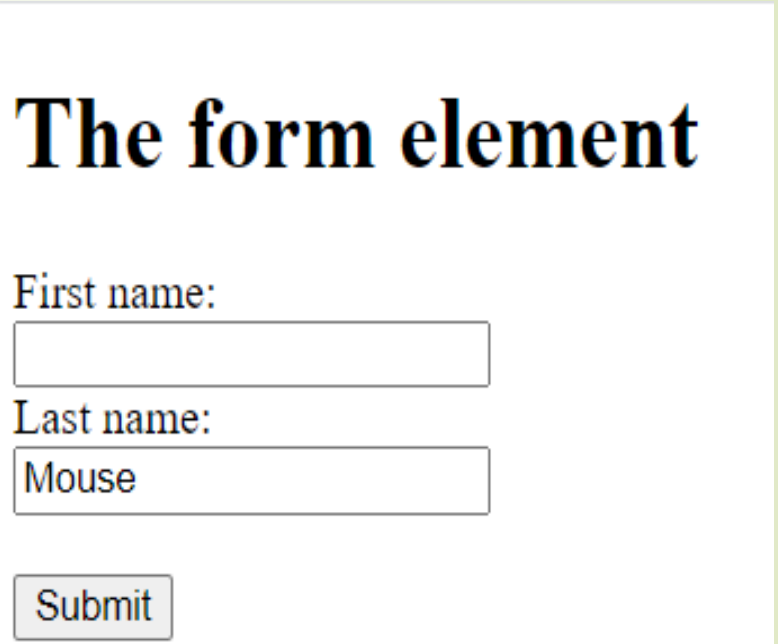
HOW FORMS WORK



The <Input> Element

- One of the most straightforward form input types is the text-entry field for entering a single word or line of text. In fact, it is the default input type, which means it is what you'll get if you forget to include the type attribute or include an unrecognized value.
- The <input> element can be displayed depending on the type attribute:
  - text: one-line text input field
  - radio: selecting one of many choices
  - submit: submit button for submitting the form
  - password: the user inputs are hidden and replaced by "*"

```
<!DOCTYPE html>
<html>
<body>

<h1>The form element</h1>

<form action="">
 First name:<br>
 <input type="text" value="">
 <br>
 Last name:<br>
 <input type="text" name="" value="Mouse">
 <br><br>
 <input type="submit" value="Submit">
 </form>
</body>
</html>
```



```
<p><input type="submit"> <input type="reset" value="Start over"></p>
```

First Name:

Last Name:

Submit Start over

Radio and Checkbox Buttons

- Both checkbox and radio buttons make it simple for your visitors to choose from a number of provided options. They are similar in that they function like little on/off switches that can be toggled by the user and are added with the input element. They serve distinct functions, however

Radio buttons

- Radio buttons are added to a form via the input element with the type attribute set to “radio.” Here is the syntax for a minimal radio button:

```
<input type="radio" name="variable" value="value">
```

- The name attribute is required and plays an important role in binding multiple radio inputs into a set. When you give a number of radio button inputs the same name value (“age” in the following example), they create a group of mutually exclusive options.

```
<!DOCTYPE html>
<html>
<body>

<h1>The form element</h1>

<p>How old are you?</p>
<ol>
 <li><input type="radio" name="age" value="under24" checked> under
24</li>
 <li><input type="radio" name="age" value="25-34"> 25 to 34</li>
 <li><input type="radio" name="age" value="35-44"> 35 to 44</li>
 <li><input type="radio" name="age" value="over45"> 45+</li>
</ol>
</body>
</html>
```

The form element

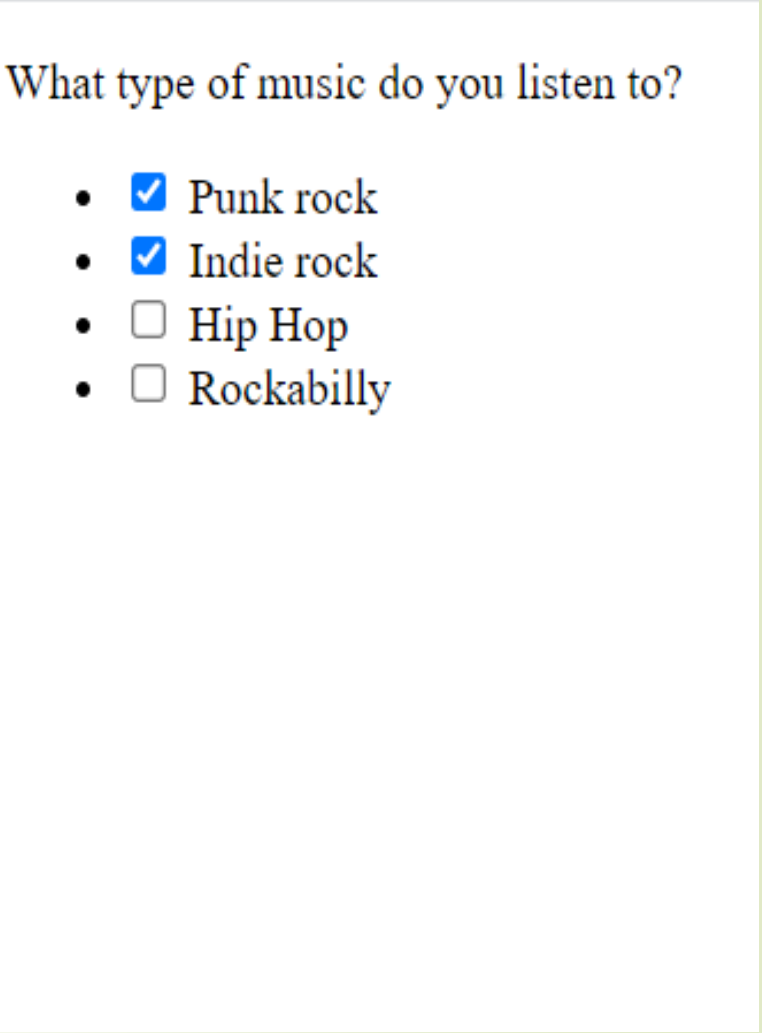
How old are you?

1. ○ under 24
2. ○ 25 to 34
3. ◉ 35 to 44
4. ○ 45+

Checkbox button

- Checkboxes are added via the input element with its type set to checkbox. As with radio buttons, you create groups of checkboxes by assigning them the same name value. The difference, as we've already noted, is that more than one checkbox may be checked at a time. The value of every checked button will be sent to the server when the form is submitted.

```
<!DOCTYPE html>
<html>
<body>
<p>What type of music do you listen to?</p>
<ul>
 <li><input type="checkbox" name="genre" value="punk" checked> Punk
rock</li>
 <li><input type="checkbox" name="genre" value="indie" checked> Indie
rock</li>
 <li><input type="checkbox" name="genre" value="hiphop"> Hip Hop</li>
 <li><input type="checkbox" name="genre" value="rockabilly">
Rockabilly</li>
</ul>

</body>
</html>
```



Checkbox buttons also use the **checked** attribute to make them preselected when the form loads

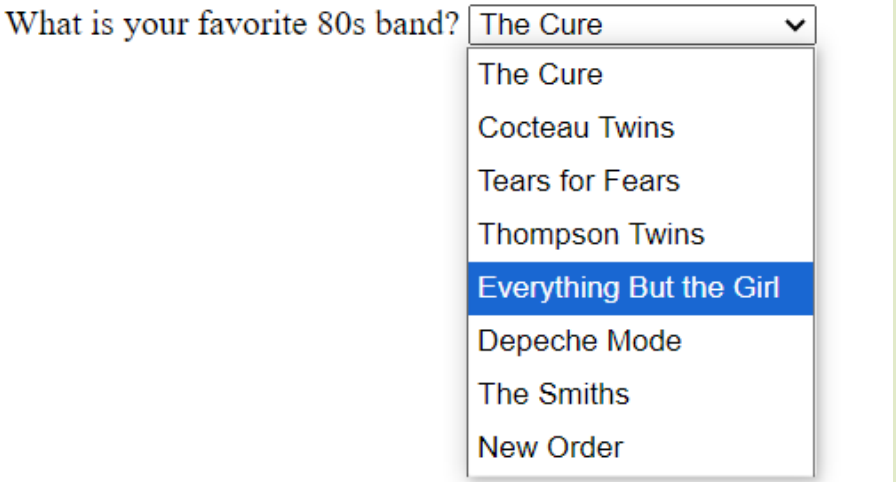
Menus

- Another way to provide a list of choices is to put them in a drop-down or scrolling menu. Menus tend to be more compact than groups of buttons and checkboxes. You add both drop-down and scrolling menus to a form with the select element. Whether the menu pulls down or scrolls is the result of how you specify its size and whether you allow more than one option to be selected.

Drop-down menus

- The select element displays as a drop-down menu (also called a pull-down menu) by default when no size is specified or if the size attribute is set to 1. In pull-down menus, only one item may be selected. Here's an example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<p>What is your favorite 80s band?
<select name="EightiesFave">
 <option>The Cure</option>
 <option>Cocteau Twins</option>
 <option>Tears for Fears</option>
 <option>Thompson Twins</option>
 <option value="EBTG">Everything But the Girl</option>
 <option>Depeche Mode</option>
 <option>The Smiths</option>
 <option>New Order</option>
</select>
</p>


</body>
</html>
```

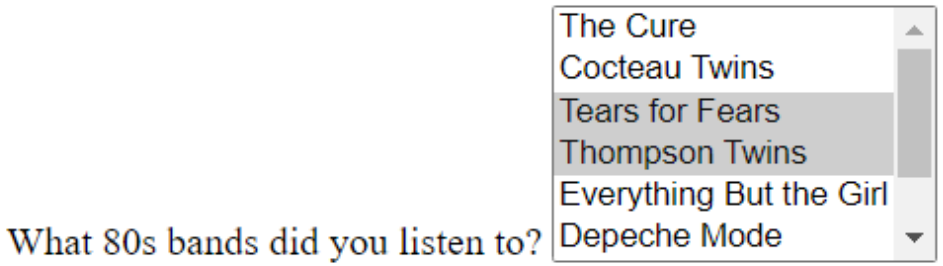


- You can see that the select element is just a container for a number of option elements. The content of the chosen option element is what gets passed to the web application when the form is submitted. If, for some reason, you want to send a different value than what appears in the menu, use the value attribute to provide an overriding value. For example, if someone selects “Everything But the Girl” from the sample menu, the form submits the value “EBTG” for the “EightiesFave” variable. For the others, the content between the option tags will be sent as the value.

Scrolling menus

To make the menu display as a scrolling list, simply specify the number of lines you'd like to be visible using the size attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<p>What 80s bands did you listen to?
<select name="EightiesBands" size="6" multiple>
 <option>The Cure</option>
 <option>Cocteau Twins</option>
 <option selected>Tears for Fears</option>
 <option selected>Thompson Twins</option>
 <option value="EBTG">Everything But the Girl</option>
 <option>Depeche Mode</option>
 <option>The Smiths</option>
 <option>New Order</option>
</select>
</p>



</body>
</html>
```



Grouping menu options

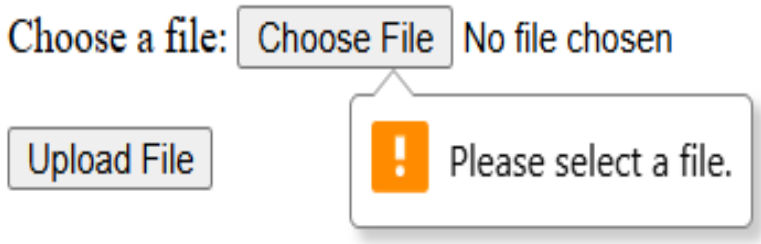
- You can use the optgroup element to create conceptual groups of options. The required label attribute provides the heading for the group:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
 <select name="icecream" size="7" multiple>
 <optgroup label="traditional">
 <option>vanilla</option>
 <option>chocolate</option>
 </optgroup>
 <optgroup label="fancy">
 <option>Super praline</option>
 <option>Nut surprise</option>
 <option>Candy corn</option>
 </optgroup>
</select>
</body>
</html>
```

File Selection Control

- Web forms can collect more than just data. They can also be used to transmit external documents from a user's hard drive. For example, a printing company could use a web form to upload artwork for a business card order. A magazine could use a form to collect digital photos for a photo contest.
- The file selection control makes it possible for users to select a document from the hard drive to be submitted with the form data.

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>File Upload Example</title>
</head>
<body>

  <h2>Upload a File</h2>

  <!-- enctype is required for file upload -->
  <form action="upload.php" method="post"
enctype="multipart/form-data">
    <label for="file">Choose a file:</label>
    <input type="file" id="file" name="myfile" required>
    <br><br>
    <input type="submit" value="Upload File">
  </form>

</body>
</html>
```



🔷 Explanation

Attribute	Description
`action="upload.php"`	The server-side script that handles the uploaded file.
`method="post"`	Must use `POST` for file uploads.
`enctype="multipart/form-data"`	**Required** to properly send file data to the server.
`<input type="file">`	Creates the file selection field.
`name="myfile"`	The name used to access the file on the server (e.g., `$_FILES['myfile']` in PHP).

html

Copy code

```html
<input type="number" id="age" name="age">
```

Here,

`id="age"` means the input element's **unique name** in the document is "**age**".

What is enctype?

html

Copy code

```html
<form action="upload.php" method="post" enctype="multipart/form-data">
```

🔷 Meaning of `multipart/form-data`

The value `multipart/form-data` means:

> "Send the form data as multiple parts — one for each input field, including files."

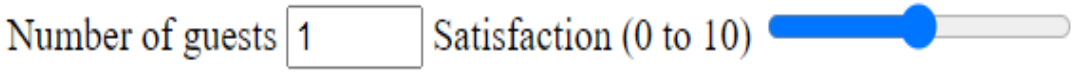
Each part contains:

- A **header** describing the field (like its name and content type)
- The **actual data** (such as text or file contents)

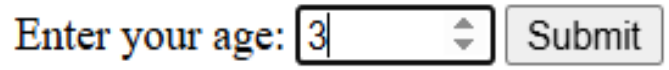
Numerical Inputs

- The **number** and range input types collect numerical data. For the number input, the browser may supply a spinner widget with up and down arrows for selecting a specific numerical value.
- The **range** input is typically displayed as a slider.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
 <label>Number of guests <input type="number" name="guests" min="1"
max="6"></label>

<label>Satisfaction (0 to 10) <input type="range" name="satisfaction"
min="0" max="10" step="1"></label>

</body>
</html>
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>File Upload Example</title>
</head>
<body>

<form>
  <label for="age">Enter your age:</label>
  <input type="number" name="age" min="1" max="100" step="1">
  <input type="submit" value="Submit">
</form>

</body>
</html>
```



🔸 Attributes:

Attribute	Description
`min`	Minimum value allowed
`max`	Maximum value allowed
`step`	Interval between legal values (e.g., 1, 0.5, 10)
`value`	Default starting value
`readonly` / `disabled`	Prevent editing / disable the field

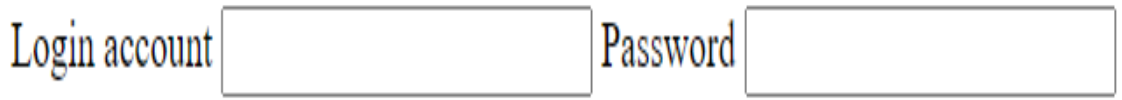
- Both the number and range input types accept the min and max attributes for specifying the minimum and maximum values allowed for the input (again, the browser could check that the user input complies with the constraint). Both min and max are optional, and you can also set one without the other. Negative values are allowed. When the element is selected, the value can be increased or decreased with the number keys on a computer keyboard, in addition to being moved with the mouse or a finger

Color Selector

- The intent of the color control type is to create a pop-up color picker for visually selecting a color value similar to those used in operating systems or image-editing programs. Values are provided in hexadecimal RGB values (#RRGGBB).

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<label>Your favorite color: <input type="color" name="favorite">
</label>

</body>
</html>
```

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<label for="form-login-username">Login account</label>
<input type="text" name="login" id="form-login-username">

<label for="form-login-password">Password</label>
<input type="password" name="password" id="form-login-password">

</body>
</html>
```



Labels

- The other method, called explicit association, matches the label with the control's id reference. The for attribute says which control the label is for. This approach is useful when the control is not directly next to its descriptive text in the source. It also offers the potential advantage of keeping the label and the control as two distinct elements, which you may find handy when aligning them with style sheets.

The <label> tag defines a **text label** for a form control

🔷 The `for` Attribute

The `for` attribute in `<label>` links it to a specific form control **using that control's** `id`.

✅ Syntax:

html

Copy code

```
<label for="control_id">Label text</label>
<input type="text" id="control_id">
```

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<form>
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
<br><br>

  <label for="password">Password:</label>
  <input type="password" id="password"
name="password"><br><br>

  <input type="submit" value="Login">
</form>

</form>


</body>
</html>
```

Username:

Password:

Login

✅ When you click the text “Username” or “Password”, the cursor automatically jumps into the matching input box.

Fieldset and legend

- The fieldset element indicates a logical group of form controls. A fieldset may also include a legend element that provides a caption for the enclosed fields.

```
<head>
<title>Page Title</title>
</head>
<body>

<form>
  <fieldset>
    <legend>Personal Information</legend>

    <label for="name">Full Name:</label>
    <input type="text" id="name" name="name"><br><br>

    <label for="age">Age:</label>
    <input type="number" id="age" name="age"><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email">
  </fieldset>

  </form>


</body>
</html>
```

```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<form>
   <fieldset>
    <legend>Preferences</legend>

    <label><input type="checkbox" name="newsletter">
Subscribe to newsletter</label><br>
    <label><input type="checkbox" name="updates">
Receive updates</label>
  </fieldset>

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


</body>
</html>
```

Text Area element

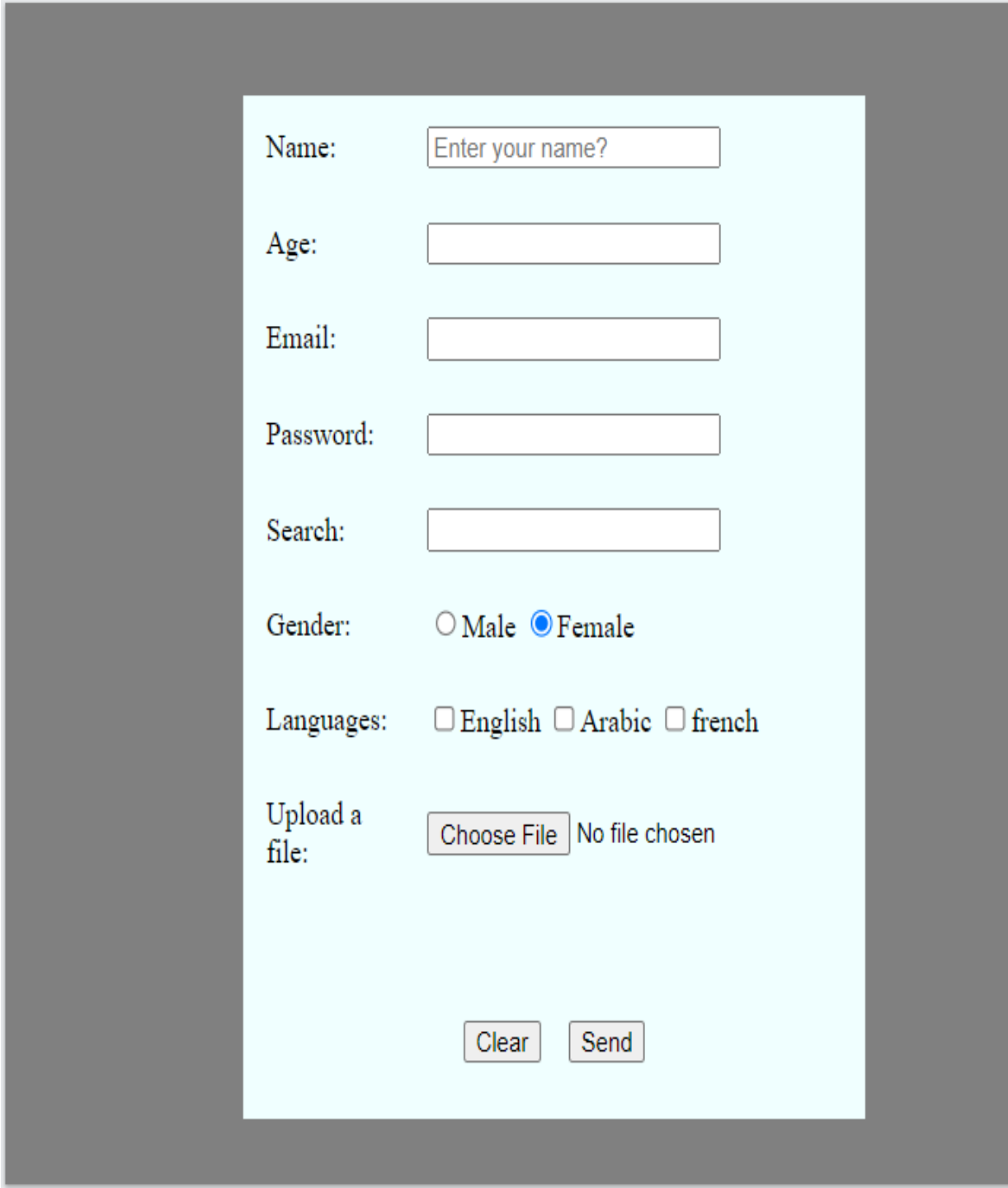
- Used to create a mutli-line text input.
- Is not an empty element
- Have an opening and a closing tag,

```
<!doctype html>
<html>
 <body bgcolor="gray">

    <form action="">
 <textarea name="txt1" cols="50" rows="30">
Enter your text...
 </textarea>
 <br>
 <input type="submit" value="Submit">
</form>

   </body>

</html>
```

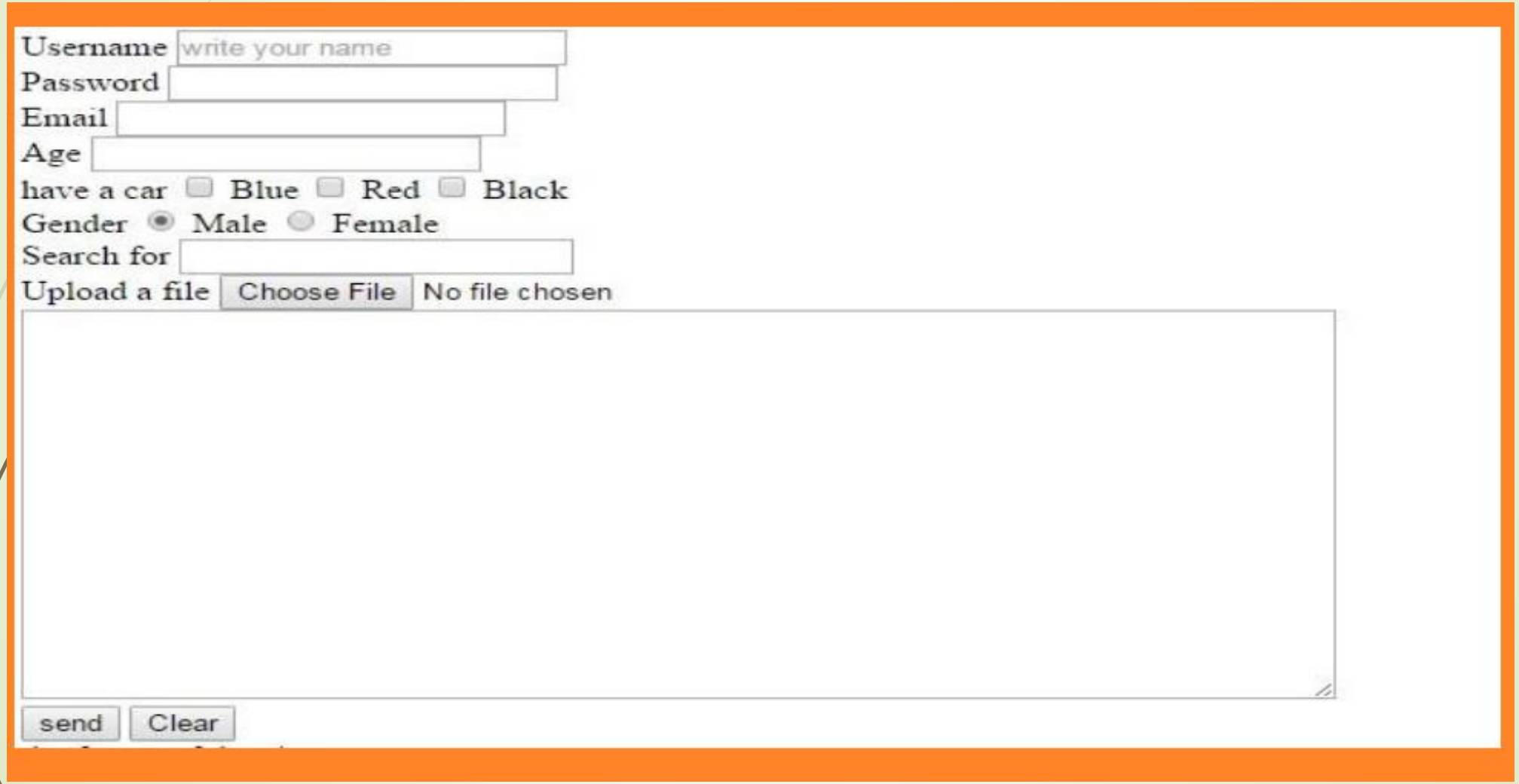
Example for A form

```
<html>
 <body bgcolor="gray">

    <table  align="center" width="50%" height="500"  cellspacing="5"
cellpadding="10" style="background-color:azure " >
   <form >
   <tr >

      <td> <label> Name:</label></td>
      <td ><input type="text" placeholder="Enter your name?"> </td>

    </tr>
      <tr>

      <td> <label> Age:</label></td>
      <td ><input type="number" > </td>

    </tr>

    <tr>

      <td> <label> Email:</label></td>
      <td  width="100%"><input type="email" > </td>

    </tr>

    <tr>

      <td> <label> Password:</label></td>
      <td ><input type="password" > </td>

    </tr>

    <tr>

      <td> <label> Search:</label></td>
      <td  ><input type="search" > </td>

    </tr>

    <tr>

      <td> <label> Gender:</label></td>
      <td><input type="radio" name="gender" checked="checked" >Male
      <input type="radio" name="gender" checked="checked" >Female
      </td>

    </tr>

   <tr>

      <td> <label> Languages:</label></td>
      <td>
       <input type="checkbox"  >English
      <input type="checkbox"  >Arabic
      <input type="checkbox"  >french
      </td>

    </tr>
```

```
    <tr>

     <td width="100"> <label> Upload a file:</label></td>
     <td >
         <input type="file" >

     </td>

    </tr>

     <br><br>

    <tr>

      <td colspan="2" style="text-align:center;font-size:50pt">
      <input type="reset"  value="Clear">
      <input type="submit"  value="Send">
     </td>

    </tr>

   </form>
   </table>

  </body>

</html>
```



Task : Design the following



References

- Jennifer Niederst Robbins, LEARNING WEB DESIGN A BEGINNER'S GUIDE TO HTML, CSS, JAVASCRIPT, AND WEB GRAPHICS, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol 2018.
- Michael Gabriel, Beginner's Guide to HTML.
- Lance Douglas Jackson, INTRODUCTION TO THE INTERNET AND WEB PAGE DESIGN, Master of Arts in Professional Communication 2009.