Session 2

Mohamed Emary

March 6, 2024

HTML Forms

HTML forms are used to collect user input. To make a form in HTML you need to use `<form>` tag. The user input is most often sent to a server for processing. The form tag is used to create an HTML form for user input.

Form can contain input elements like text fields and labels for these fields.

Example:

```html
<form action="/action_page.php">
  <label for="fname">First name:</label><br/>
  <input type="text" id="fname" name="fname"><br/>
  <label for="lname">Last name:</label><br/>
  <input type="text" id="lname" name="lname"><br/><br/>
  <input type="submit" value="Submit">
</form>
```

As you see in the example the value in `for` in the `label` element should be exactly the same as the value in `id` of the `input` field.

Each input field should hava a `name` attribute to be able to be sent to the server.

Type Attribute

Sometimes you want the user to enter a password, an email, a phone number, a date, a color, a number, etc. HTML5 has a lot of new input types for these cases. Just use the `type` attribute to specify the type of input.

There is a lot of input types, some of them are:

1. `text` (*default*)
2. `password`
3. `email`
4. `number`
5. `date`
6. `color`
7. `url`
8. `tel`
9. `search`
10. `file`
11. `radio`
12. `checkbox`

Example:

```html
<form>
  <label for="email">Email:</label><br/>
  <input type="email" id="email" name="email"><br/>
  <label for="password">Password:</label><br/>
  <input type="password" id="password" name="password"><br/><br/>
  <input type="submit" value="Submit">
</form>
```

Each input element should have a `name`, `type` attributes.

You can also add the `value` attribute to set the default value of the input field.

The submit button is used to send the form to the server and it has a `submit` type.

The `<br/>` used in the example above is used to add a line break between the input fields.

To allow uploading files you can use the `file` type, and to allow multiple files you can use the `multiple` attribute.

You can also allow only a specific type of files using the `accept` attribute.

Example:

```html
<form>
  <label for="file">Select files:</label><br/>
  <input type="file" id="file" name="file" multiple
   ↪ accept=".png"><br/>
  <input type="submit">
</form>
```

To allow any type of images you can use `accept="image/*"`. See this link for more information.

Button

The `button` element has a `type` attribute that can be `submit`, `reset`, or `button`.

- `submit`: The button submits the form data to the server. (This is the *default*)
- `reset`: The button resets all the form data to its initial values.
- `button`: The button does nothing.

Radio Buttons & Checkboxes

You can also use a regular `input` tag instead of `button` and set the `type` attribute to `submit`, `reset`, or `button`, but it's better to use the `button` tag.

Another type of input we can use is `radio`. One important thing about `radio` is that all the radio buttons in a group should have the same `name` attribute. You should also add `value` attribute to each radio button.

When you add a label for each radio button, you can click on the label to select the radio button.

Example:

```html
<form>
  <fieldset>
    <legend>Select your favorite fruit:</legend>

    <input type="radio" id="apple" name="fruit" value="apple">
    <label for="apple">Apple</label><br>

    <input type="radio" id="banana" name="fruit" value="banana">
    <label for="banana">Banana</label><br>

    <input type="radio" id="cherry" name="fruit" value="cherry">
    <label for="cherry">Cherry</label>
  </fieldset>
</form>
```

The same applies for checkboxes except that you can select multiple checkboxes at the same time.

Textarea

The `textarea` tag is used to create a multi-line text input. The `rows` and `cols` attributes are used to specify the size of the textarea but the user can still resize it using the mouse.

Example:

```
<form>
  <label for="message">Message:</label><br/>
  <textarea id="message" name="message" rows="4" cols="30"></textarea><br/>
  <input type="submit">
</form>
```

Select Option

The `select` tag is used to create a drop-down list and the `option` tag is used to define the options in the list. If you don't give a `value` attribute to the `option` tag, the value will be the text inside the `option` tag.

You can also group the options using the `optgroup` tag, and you can only choose what is available in the list (you can't enter a value that is not in the list).

Example:

```
<form>
  <label for="cars">Choose a car:</label><br/>
  <select id="cars" name="cars">
    <optgroup label="Swedish Cars">
      <option value="volvo">Volvo</option>
      <option value="saab">Saab</option>
    </optgroup>
    <optgroup label="German Cars">
      <option value="mercedes">Mercedes</option>
      <option value="audi">Audi</option>
    </optgroup>
  </select><br/>
  <input type="submit">
</form>
```

Datalist

The `datalist` tag is used to create a list of options for the `input` tag. The `list` should take the `id` of the `datalist` tag.

It also allows the user to enter a value that is not in the list, But you can't group the options like in the `select` tag.

Example:

```
<form>
  <label for="browser">Choose a browser:</label><br/>
  <input list="browsers" id="browser" name="browser">
  <datalist id="browsers">
    <option>Chrome</option>
    <option>Firefox</option>
    <option>Internet Explorer</option>
    <option>Opera</option>
  </datalist><br/>
  <input type="submit">
</form>
```

Input Validation

In text fields you can use `maxlength` and `minlength` attributes to specify the maximum and minimum number of characters allowed in the input field.

In numbers fields you can use `max` and `min` attributes to specify the maximum and minimum values allowed in the input field.

You can also use the `required` attribute to make the input field required, so the user can't submit the form without filling this field.

In `select` if the user didn't choose any option, the first option will be sent to the server, and to change that you can use the `selected` attribute in the `option` tag you want to be the default.

To hide an input field from the user you can use the `hidden` attribute. And to disable an input field you can use the `disabled` attribute. The disabled input fields doesn't send any data to the server however the hidden input fields do.

To prevent the user from changing the value of an input field you can use the `readonly` attribute.

Table 1: hidden, disabled, readonly attributes

Attribute	Description
`hidden`	Hides the input field from the user but sends the data to the server.
`disabled`	Disables the input field and doesn't send the data to the server.
`readonly`	Prevents the user from changing the value of the input field, and sends the data to the server.

Extra Information

To check wheather a tag or attribute you are using is supported by the browser you can use the caniuse website.

To check your HTML code markup you can use W3C Markup Validation Service.