COURSEWARE

Forms

Contents

Overview

Forms in HTML are used to collect user input, which can then be sent to another page or sent to a server for processing.

We create a form by using `<form>` and `</form>` tags, nesting our form elements within it. Within the first `<form>` tag we also specify the `action` and the `method`:

- **Action** - URL to open / action to execute upon form submission
- **Method** - How the information is passed to the server:
  - `GET`: Appends the arguments to the `action` URL and opens it as if it were an anchor
  - `POST`: Posts the information to whichever URL the form points to

```
<form method="post" action="scripts/subscribe.pl">
  <!-- Some elements -->
</form>
```

Form Options

Text Input

Textual input is the most common form element, which allows us to enter text into a field:

```
<form method="post" action="scripts/subscribe.pl">
  Please enter your name:
  <input type="text" name="UserName"/>
</form>
```

The `placeholder` attribute assigns temporary text to the field, which will be wiped away when the user starts typing:

```
<form method="post" action="scripts/subscribe.pl">
  Please enter your name:
  <input type="text" placeholder="Enter Username" name="UserName"/>
</form>
```

The `value` attribute assigns default information to the field, which might be replaced by the user:

```
<form method="post" action="scripts/subscribe.pl">
  Please enter your name:
  <input type="text" value="John_Doe123" name="Username"/>
</form>
```

Buttons

Buttons allow us to interact with forms, though there are three distinct types of thing which a user might call a *button*:

- `submit`: sends information in the form using whichever `action` is defined in the `<form>` tag
- `reset`: sets the entire form to its initial state, using default settings if assigned
- `button`: acts as a trigger for any client-side script assigned to it

In the below example, the `submit` button will call the `subscribe.pl` script defined in the `action` attribute of the `<form>`, the `reset` button will clear the form, and the standard `button` will do nothing:

```
<form method="post" action="scripts/subscribe.pl">
  Please enter your name:
  <input type="text" name="UsrName"/>
  <br/>
  <button type="submit" name="OKBtn" value="OK" />
  <button type="reset" name="ResetBtn" value="Reset" />
  <button type="button" >Click me</button>
</form>
```

Multi-line text input

Multi-line text input, such as for feedback and contact forms, use the `<textarea>` tag, which contains some handy attributes:

- `rows` - attribute specifies the visible number of lines in a text area.
- `cols` - attribute specifies the visible width of a text area.
- The text between `<textarea>....</textarea>` is the default text that will show in the text area.

```
Address:
<textarea name="UsrAddr" rows="7" cols="24" >
  Enter address here
</textarea>
```

Multiple-choice selection

HTML is perfectly capable of handling multiple-choice selection:

- *Checkboxes*: allows for multiple selections from a group placed at the same scope
- *Radio buttons*: allows for single selection from a group placed at the same scope

(*note that radio buttons must all have the same* `name` *attribute*):

```
<form ...>>
  <input type="radio" name="RadioDrink" value="Tea"/>Tea
  <input type="radio" name="RadioDrink" value="Coffee"/>Coffee
  <input type="radio" name="RadioDrink" value="Soup"/>Soup
  <br/>
  <input type="checkbox" name="CheckMilk" value="Yes"/>Milk
  <input type="checkbox" name="CheckSugar" value="Yes"/>Sugar
  <br/>
  <input type="submit" name="OKButton" value="Make Order"/>
</form>
```

`<select>` and `<option>`

`<select>` and `<option>` are used to create a drop-down field with predefined values.

Multiple selection may be achieved by using the optional `multiple` attribute, which is itself optionally limited with the `size` attribute, as shown below:

```
<form ...>
<select id="cars" name="cars" size="2" multiple>
    <option value="audi">Audi</option>
    <option value="bmw" selected>BMW</option>
    <option value="mercedes">Mercedes</option>
  </select>
</form>
```

By default, the *first* value in the list is selected, but can be overridden with the `selected` attribute - above, the default value is "BMW".

`<datalist>`

A `<datalist>` is similar to a `<select>` list, but a `<datalist>` allows you to type the option you wish into the drop-down field. The autocompletion is somewhat buggy, however, so use this sparingly.

```
<form ...>
  <input list="browsers" />
  <datalist id="browsers">
        <option value="Internet Explorer">
        <option value="Firefox">
        <option value="Google Chrome">
        <option value="Opera">
        <option value="Safari">
  </datalist>
</form>
```

Email / URL

The `email` and `url` `<input>` types both use automatic *syntax verification*; upon beginning to type into either field, they will begin to automatically validate whether valid email/URL syntax has been entered or not:

```
<form ...>
 <input type="email" name="email"/>
 <input type="url" name="url"/>
</form>
```

Number

The `number` type allows for incrementing or decrementing of a predefined scale, which can be set a default `value`.

```
<form ...>
 <input type="number" min="1" max="100" step="2" name="phoneNumber"/>
</form>
```

This also automatically validates entered text to check that it matches numbers.

Range

The `range` type utilises a slider to select a particular number, which can also be set a default `value`.

```
<form ...>
 <input type="range" min="1" max="10" name="ourService" value="9"/>
</form>
```

(*note: This works differently in different browsers.*)

Date

The `date` picker type utilises a calendar, and allows for the selection of a certain date:

```
<form ...>
 <input type="date" name="myBirthday"/>
</form>
```

Search

The `search` type provides a semantic definition for search input.

There should never be more then *one* search field on a Web page. We also need to set the name for the search field, otherwise nothing will be submitted.

The most common name is q:

```
<form ...>
 <input type="search" q="searchProducts"/>
</form>
```

Colour input type

The `colour` input type lets us select a colour from a predefined list or from specifying an RGB / HSL value:

```
<form ...>
 <input type="colour" name="myFavColour"/>
</form>
```

Pattern

The `pattern` attribute can be used to in order to implement REGEX patterns to a particular field for validation e.g. check a debit card number:

```
<form ...>
 <input type="text" pattern="[0-9]{13,16}" name="creditCardNumber"/>
</form>
```

The `pattern` attribute can be used with the following input types:

- `text`
- `search`
- `url`
- `tel`
- `email`
- `password`

Required and autofocus

The `required` attribute forces a field to be mandatory on the client-side.

Generally it is paired with `autofocus`, which automatically moves the cursor focus to a particular field:

```
<form ...>
    <input type="text" autofocus="true" required />
</form>
```

Fieldset

A `<fieldset>` is used to group related form elements together, and a meaningful legend provides accessibility by splitting sections e.g. user details, address details.

```
<fieldset>
    <legend>Your details</legend>
    <label for="fname">First name</label>
    <input id="fname" type="text">
    <label for="sname">First name</label>
    <input id="sname" type="text">
    <label for="age">Age</label>
    <input id="age" type="number">
</fieldset>
```

Tutorial

There is no tutorial for this module.

Exercises

1. Create form with the following criteria:
    - Action to redirect to "index.html"
    - Method "post"
    - `h1` with the value of "Sign up form"
    - Legend with the value of "Your basic info"
    - Name, Email and Password input fields
    - Gender radio selection
    - Button with the value of "Sign Up!"

Image of form

▼ Solution

```
<!Doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up Form</title>
</head>
<body>
    <form action="index.html" method="post">
        <h1> Sign Up </h1>
        <fieldset>
            <legend><span class="number">1</span> Your Basic Info</legend>
            <label for="name">Name:</label>
            <input type="text" id="name" name="user_name">
            <br>
            <label for="email">Email:</label>
            <input type="email" id="mail" name="user_email">
            <br>
            <label for="password">Password:</label>
            <input type="password" id="password"       name="user_password">
            <br>
            <label>Gender:</label>
            <br>
            <input type="radio" id="m_gender" value="m_gender" name="gender">
<label for="m_gender">Male</label><br>
            <input type="radio" id="f_gender" value="f_gender" name="gender">
<label for="f_gender">Female</label>
        </fieldset>
        <button type="submit">Sign Up</button>
    </form>
</body>
</html>
```