HTML Forms and Media Tags

Ever wondered how all these websites collect user data and provide relevant information after processing it? It is done using HTML forms and they're pretty simple to learn.

- Forms are a crucial part of web development as they act as a link between the front-end and the backend part of the website.

FORMS

With the help of forms, a user enters the data then

- This data is either processed by the browser itself (using Javascript) or the data goes to the backend servers where it gets processed.
- A **form** contains form elements. It is defined with the **<form>** tag.
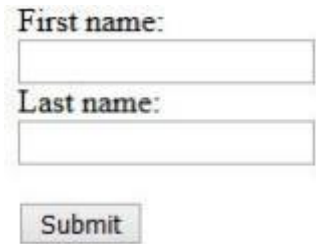
An HTML form can contain elements like text fields ex. single-line/multiline, select boxes, checkboxes, buttons and radio buttons etc. We will start with just a basic form i.e. without any elements:

Eg:

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

```
</form>
```

The form on your page will look like this:

First name:

Last name:

Submit

Now, we will discuss the tags and attributes we mainly use in forms.

input Tag

- In the **<input>** tag, users enter the data. This is called an **inline** tag.

1. **type Attribute:** HTML gives many types of input which could be used for distinct kinds of entries. By default, the type value is **"text"**, which signifies that single-line text input is required. Some of the values for the type attribute are as follows:

   E.g <input type = "email">

Type	Description
email	It is used for input fields that should contain an email address and has validation parameters.
password	A single-line text field that defines a password field whose value is hidden.
date	It is used for input fields that contain the date(year, month, and day with no time)
number	It is used for numeric input fields.
range	It defines a control for entering a number whose exact value is not important. You can set restrictions on what numbers are accepted with min, max, and step attributes.
url	It is used to let the user enter the URL.
radio	It defines a radio button that allows a single value to be selected out of multiple choices.
hidden	It defines a hidden input field that is not displayed but whose value is submitted to the server.
time	It allows the user to select a time with no time zone.

file	It allows the user to select the file and Browse button for file uploads.

Here **type="submit"** signifies a button which when selected, will submit the form. We can also control the text appearing on the submit button by using the value attribute. Ex: <input type="submit" value="Submit">, would make a button like this -
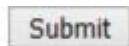
Submit

NOTE: *The type attribute is mandatory.*

EXTRA:

You can also refer other types from this link:

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

2. **value Attribute:** Value attribute is really useful, although it is not a compulsory attribute to add to an input element.

    a. For "button", "reset", and "submit" - This attribute defines the text on the button.

    b. For "text", "password", and "hidden" - This attribute defines the initial (default) value of the input field.

Eg. to display country as the default initial value in an input field, the code will be like this:

<input type=" text" value=" India" />

This will be shown on the screen like

India

3. **name Attribute:** Without the **name** attribute, this form element won't be submitted or in other words would not be sent to the server. So, this is a compulsory attribute for input tags in a form.

   The **input value is accessed using the name attribute** and it also uniquely identifies that piece of data. For eg:

   ```html
   <form action="#">
       <input type="text" name="abc">
   </form>
   ```

   The text entered in the input field will be referenced using the value of the *name* attribute which is **abc** in this case.

label Tag

- The label tag is used to describe the kind of input in a form. This is not a compulsory tag.
- You can also describe the kind of input format without the use of a label tag, as also shown in the above example. But it is always best to use **the <label>** tag.
- **<label>** tag can also be tied to their form elements like <input>, <textarea>, etc. "label" also an *inline tag.*

   Like what we have done in the above example is:

   ```html
   <label for="fname">First name:</label><br/>
   <input type="text" id="fname" name="firstname" />
   ```

   The label is tied to this input element by giving the **"id"** attribute of the input element the same value as the label's **"for"** attribute.

NOTE: *It is possible for the value of id and name to be the same and most of the time this will be the case.*

required Attribute

- This attribute is used to specify that an input field must be filled out before submitting the form. Otherwise, it shows a pop-up to fill out the required field.,

- This attribute is a **boolean attribute.**

- **If a boolean attribute is present,** Its value is **true,** and if it's absent, its value is false.

    Eg. We will apply a **required** attribute to an input field:

```
<form>
    <input type="text" required>
    <input type="submit" value="Submit" />
</form>
```

will show an error, when clicked on Submit button as shown in the image below:

placeholder Attribute

- The **placeholder** attribute can be used along with the input element. It adds a hint about the value the user will enter and this placeholder text will disappear when the user starts entering the value.

    Eg. adding a placeholder to the input:

```
<input type="text" name="fname" placeholder="First name"><br>
<input type="text" name="lname" placeholder="Last name">
```
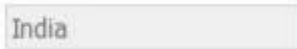
will make the input look like this:

disabled Attribute

- If we want to disable the <input> element then the disabled attribute is used. By disabling the input, it becomes uneditable and unclickable, although it might already contain a default value in it.
- This is also a **boolean attribute.**

Eg. adding a disabled attribute to an input element,

```
<input type="text" name="country" value="India" disabled><br>
```
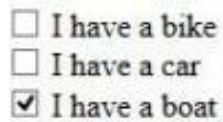
will make the input look like this:

```
India
```

NOTE: *Disabled elements in a form will not be submitted.*

Checkboxes
- The **checkbox** is a value of the **type attribute** of the input element.
- Whenever **more than one option may need to be checked** checkboxes are used. You can also use checkboxes to enable or disable something.
- Whenever a checkbox is activated, it is shown as a square box that is ticked (checked).
- If you want the checkbox selected by default when the page loads, that there is an attribute named **checked**

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car<br>
  <input type="checkbox" name="vehicle3" value="Boat" checked> I have a boat
</form>
```

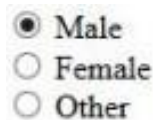The above code will display checkboxes like the below image:

Radio Button

- A radio button is almost similar to a checkbox, but the values of the name attributes are all the same in a radio button. The value of type attribute is **"radio"** to define a radio button.
- As we want to **only select one of the radio buttons at once,** the name attributes are all set to the same value to make these radio buttons part of the same set.

Eg:

```
<form action="/action_page.php">
    <input type="radio" name="gender" value="male" checked> Male<br>
    <input type="radio" name="gender" value="female"> Female<br>
    <input type="radio" name="gender" value="other"> Other
</form>
```

The above HTML code shows like

fieldset and legend Elements

- The **<fieldset>** element is used to provide grouping for a part of an HTML form. To make the elements more presentable, the <fieldset> tag draws a box around the related elements.
- There is a <legend> tag that comes just after the <fieldset> tag. To provide a title or explanatory caption for the rest of the contents of the legend element's parent element, this **<legend>** element is used

Eg: if a fieldset and legend is used:

```html
<form>
    <fieldset>
        <legend>SUBSCRIBE:</legend>
        Name: <input type="text"><br> Email: <input type="text"><br>
    </fieldset>
</form>
```

The form would look like this:

```
┌─SUBSCRIBE:─────────────────────┐
│ Name: [                      ] │
│ Email: [                     ] │
│                                │
└────────────────────────────────┘
```

select Element

- HTML **<select>** tag is used to create a drop-down list of options. There can be many options in a dropdown list and the user can choose one of them.

- To represent the associated data submitted to the server, the select tag also contains a name attribute, like other form elements.

There are some of the unique attributes of the select element-

- **multiple**, which helps to specifies that multiple options can be selected
- **size**, which specifies how many options can be shown at once.

Eg:

```html
<select name="select">
    <option value="value1">Value 1</option>
    <option value="value2" selected>Value 2</option>
    <option value="value3">Value 3</option>
</select>
```
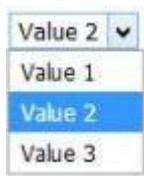
it will show a dropdown like this:

Value 2 ∨

and clicking the above option will open other options as

Value 2 ∨
Value 1
Value 2
Value 3

option Element

- To define the possible options, the **<option>** tag is used. This tag is put inside the **<select>** tag.
- A separate <option> element is used, for every option in the drop-down list.

By default, the first <option> element from the options' list is selected. To change this predefined option, we need to use the **selected** attribute with the <option> tag.

The value attribute should be there in each option element, which contains the data value that will be submitted to the server when that option is selected.

optgroup Element

- The **<optgroup>** tag will create separate groups of options inside the dropdown.
- It is used to group several options together into one group.

```
<select>
    <optgroup label="Books">
    <option value="html">HTML</option>
    <option value="css">CSS</option>
    </optgroup>
    <optgroup label="Snippets">
    <option value="git">Git</option>
    <option value="java">Java</option>
    </optgroup>
</select>
```

This will show the dropdown list as:

textarea Element

- The **<textarea>** element is an input element where the user can input multi-line text, unlike the <input> element where there is only a single line.
- A text area can hold an **unlimited number** of characters, and **text wrapping** is allowed when the form is submitted.

Eg:

```
<label>Write about yourself:</label>
<textarea rows="10" cols="50"></textarea>
```

Will show something like this:

-
- **rows and cols Attribute**

These 2 attributes are used to set the size of <textarea>.

- The **rows** attribute specifies the visible height of a textarea.

- The **cols** attribute specifies the visible width of a text area.

NOTE: The size of the text area can also be specified by CSS height and width property.

Submit Button

- **Submit button** is a button that is used to **automatically submit the form** when clicked.

- This button is present at the end of the form.

To add the submit button to the form. there are 2 different ways:

- **via <input> tag**

- **via <button> tag**

Both these ways will work in the same way.

- **via input tag**

- The **<input>** tag can also be used to create a button.

- To use it as a button, the **type attribute** is set to value submit.
When the click event occurs, i.e. the user clicks on the button, the form gets submitted. The input tag is a self-closing tag, so the value of the button is set by the **value attribute.**

The submit button can be formed using the input tag as

```
<input type="button" value="Submit form">
```
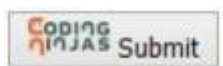
this will display the button as:

- **via button tag**
  - The **<button>** is also used to create a submit button in the form.
  - Although the <input> tag also creates a submit button, there are some benefits of the button tag over the input tag.
  - The button tag is a container tag and therefore, can contain other tags. This helps in adding images and other content in the button.

The button has to set the type attribute to submit value, to make it a submit button.

Eg., the submit button with an image can be defined as:

```
<button type="submit">
    <img src="images/logo.png" alt="Learn HTML" height="20"
    width="40"/> Submit
</button>
```

this will display the button as:

- **autofocus Attribute**

  - The **autofocus** attribute is a boolean attribute.
  - When applied to the button specifies that the button automatically gets focus when the page loads.

- **events on the button**

- There are some events of HTML that we will cover here.
  - Events are things that take place when something happens with HTML elements.
  - **onclick** is one such event that triggers some action when the button is clicked.

Eg:

```
<button onclick="alert('Learn HTML')" >Click ME</button>
```

the **alert** is a window method that displays an **alert box with a specified message** and an OK button. The alert box covers the whole window screen and makes that browser unavailable to use.

The alert box looks like this:
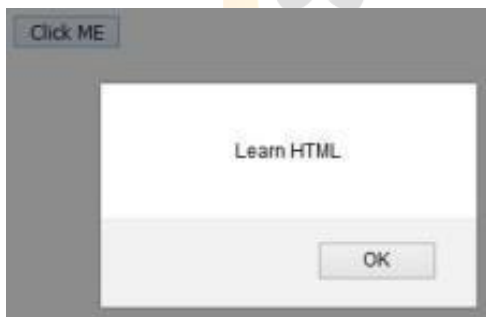
Another such window method is **location. href,** that creates an HTML **button that acts as a link**. So, when it is clicked, it redirects to a page.

The value of location. href contains the URL you want to redirect it to. Eg:

```
<button onclick="location.href='http://google.com';">Click ME</button>
```

Submitting the form

When the form is submitted, the page gets reloaded and we know that the form gets submitted. But actually, the form input data is not being submitted to the server.

We need to set 2 attributes in the form, to get the form to send the input data to the server.

- Method attribute
- Action attribute

Eg., the form has these attributes as

```
<form action="/address_to_handle_form" method="post"> </form>
```

- **method Attribute**

  The **method** attribute defines how the form data is sent. The data can be

  sent in different ways to the server. There are mainly 2 values we use to

  send the data:

  - **get -** this appends the data into the url with '?' as a separator in name-value pairs. Since this data will be visible, sensitive data (like passwords) should not be sent. This can be used to send query strings like *URL?name=value&name=value*
  - **post** - this appends the data inside the body of the HTTP request. The post is used to send sensitive data.

- **action Attribute**
  - The **action** attribute defines where the form data is sent when the form is submitted. This contains the address (i.e. URL) of the file where the data is sent. The URL can be provided in absolute and relative paths.
  - The **absolute URL** points to another website.

    (like action="http://www.xyz.com/example.html").

    The **relative URL** points to a file within the

website

    (like action="/example.html").

MEDIA ELEMENTS

Multimedia is represented in multiple forms(eg., video, audio) and in multiple formats(eg., mp3, .avi). It is a different content than text that uses sound, videos, music, animations, movies, etc. on the web. With the help of HTML media elements, we can add audio, video, and figures.

Audio and Video Element

Audios and videos can be directly embedded on a web page without any external support using HTML.

- To **add audio** to a **web page,** the **<audio>** element is used**.**
- To **add a video** to **the web page**, the **<video>** element is used**.**

But only these elements are not sufficient to add to the media. We also need to control the media as well. So to fully add the media, there are several tags and attributes that are required.

Only the tag name is different for the <audio> and <video>. They are the same in the way the content is added to it.

Eg., to add audio to the web page, we use the following code

```
<audio controls>
    <source src="cn.avi" type="audio/avi">
    <source src="cn.mp3" type="audio/mpeg">
    Your browser does not support the audio element.
</audio>
```

Here, multiple source tags are used so that the audio plays if any one of the formats

is supported by the browser. Else, the text message will be shown

The audio player will be shown like this:

- **Controls attribute**

  The controls attribute is necessary to add controls like play, pause, and

  volume to the audio/video. This gives you the ability to control the video

  and audio content**.**

- **Source tag**

  To serve the same media content in multiple formats, the <source> element is used,

  so that different browsers can run any of the files that it supports. It is an empty

  element.

- **src Attribute**

  To specify the URL of the media file that is needed to be played, the src attribute is

  used. This can have an absolute or relative path.

- **type Attribute**

  The type attribute is used to specify the media type of the media resource. The way

  we define a type for video is like video/mp4, etc., and for audio

  like: audio/MPEG.

figure Element

The **<figure>** is a new tag introduced in HTML5. This element specifies self-contained

content, like illustrations, diagrams, photos, code listings, etc.

Eg:
```html
<figure>
    <img src="images/logo.png" alt="The Pulpit Rock" width="304" height="228">
    <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>
```

The output would be:

Fig 1. Coding Ninjas Logo

- **figcaption Element**

  The <figcaption> tag defines a caption for an element. This can be placed anywhere inside the figure element.

FAVICON

- A **favicon** is a small, iconic image that represents the website.
- They are most often found in the address bar of your web browser, but they can also be used in lists of bookmarks in web browsers and feed aggregators.

You can see an icon beside the title of the page on the tab itself. This is known as a favicon.

You can add a favicon with the following syntax:

```
<link rel="icon" href="/favicon.ico" type="image/x-icon" />
```

- The **rel** defines the *relationship with the favicon.*
- The **href** defines the *location of the favicon.*
- The **type** defines the *media type of the favicon*

Eg., the icon on the tab is the favicon and looks like this:

META TAGS

- Information about the web page is stored under the meta tag as metadata defines information about data. The data stored in it is known as metadata.
- The **meta** tag is a **self-closing tag**.
- **Metadata will not be displayed on the page** but will be machine parsable.
- To specify page description, author of the document, keywords, last modified, and other metadata; meta elements are typically used.

There are 4 attributes that are used in the meta tag:

- Name

- http-equiv

- Charset

- Content

To provide information about your site to search engines, meta tags have been one of the most basic elements of SEO. **Search engine optimization (SEO)** is defined as the process of affecting the online visibility of a website or a web page in the result of a web search engine.

It's very crucial to add meta tags to your web pages. Search engines such as Google generally show the meta description in search results, they can **highly affect user visits to the website**.

NOTE: There can be *any number of meta tags* defined within a page inside the head.

Name Attribute

To specify the name for the metadata, the **name** attribute is used. This attribute is used along with the content attribute. The name attribute is used to specify a name for the information/value of the content attribute.

The name attribute can have one of these 6 values:

- keywords - It is used to specify a comma-separated list of words for SEO purposes.
- author -It is used to specify the author name of the document
- generator -It is used to specify the software packages used that is used to generate the document
- viewport - It is used to specify the control of the viewport on different devices
- description - It is used to specify the description of the page
- application-name -It is used to specify the name of the application that the page is representing.

The syntax is: <meta name="value">

NOTE: *If the http-equiv attribute is set, the name attribute should not be set. SEO is used by the search engines like google and bing to search for the website's content relevant to the user search. This increases the quality and quantity of traffic on one's website.*

Content Attribute

This attribute is used to specify the value associated with the http-equiv or name attribute.

The syntax is: <meta http-equiv/content="value" content="text">

Charset attribute

- The **charset** attribute is used for **announcing the character encoding** for a page.
- It is a good practice to use **UTF-8 encoding**. However, this must be taken into account, the declared character set matches the one present on the page and is defined for every page of the website.

http-equiv Attribute

- The **http-equiv** attribute provides an **HTTP header for the information/value of the content attribute.** The value of this attribute can be used to alter servers and user-agents behaviour.

The syntax is: `<meta http-equiv="content-type|default-style|refresh">`

Eg., the "**refresh**" value is used to specify the seconds after which the page is going to be refreshed. And if along with the time, a url is mentioned as *'5;url=https://www.codingninjas.in/',* then after 5 seconds, the user would be redirected to the mentioned URL.