Topic : HTML Forms and Media Tags

1. FORMS

HTML Forms are used to collect different kinds of user input. Through these forms, a user enters the data which is either processed by the browser itself (using Javascript) or the data goes to the servers where it gets processed.

A **form** is an area that can contain form elements. A form is defined with the **<form>** tag. **Form elements** are elements that allow the user to enter information (like text fields, textarea fields, drop- down menus, radio buttons, checkboxes, etc.) in a form.

An HTML form may contain a lot of elements like text fields (single line or multiline), select boxes, buttons, checkboxes, or radio buttons. Let us start with just a basic form without any elements:

Eg:

```
<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 about the tags and attributes we mainly use in forms.

1.1. input Tag

The **<input>** tag specifies an input field where the user can enter data.
<input> elements are used within a <form> element to declare input controls that allow users to input data.
This is an ***inline tag***.

1.1.1. type Attribute

HTML provides different types of input that you can use for different types of entries. By default the value of type is **"text"**, which specifies that we want single line text input.

Some more values for **type** attribute are-

email	password	date	number
range	url	radio	hidden
time	range		

type="submit" represents a button that when selected will submit the form. You can control the text that appears on the submit button with the value attribute. Eg: `<input type="submit" value="Submit">`, will create a button like this -

NOTE: *type attribute is mandatory.*

EXTRA:

You can see other types from this link:
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

1.1.2. value Attribute

Value is not a compulsory attribute to add to input element, but it is very useful. The value attribute is used differently for different input types:

- For "button", "reset", and "submit" - it defines the text on the button
- For "text", "password", and "hidden" - it defines the initial (default) value of the input field
- For "checkbox", "radio", "image" - it defines the value associated with the input (this is also the value that is sent on submit)

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

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

This will shown on the screen like:

1.1.3. name Attribute

The **name** attribute is a compulsory attribute for input tag in a form. Without this attribute, this form element won't be submitted or in other words would not be send to the server.

The name attribute also uniquely identifies that piece of data. **The value of the input is accessed using the name attribute**.

1.2. label Tag

Use of labels to describe the kind of input in a form is not compulsory. You can do that without the use of label tag, as also shown in above example.
But it is best to use **<label>** tag to describe the kind of input for form element. It not only is semantically correct but it can also be tied to their form elements like <input>, <textarea>, etc.

This is also an ***inline tag***.

Like what we have done in the above example is:

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

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

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

1.3. required Attribute

When present, it specifies that an input field must be filled out before submitting the form. Else, it shows a pop up to fill out the required field.
The required attribute is a **boolean attribute**.

Eg. applying **required** attribute to an input field like

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

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

1.4. placeholder Attribute

The **placeholder** attribute is used with input element. It describes a sample value or a short description of the expected format.
The value of the placeholder attribute specifies a short hint that describes the expected value of an input field.

Eg. adding 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:

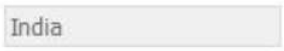
1.5. disabled Attribute

The **disabled** attribute specifies that the <input> element should be disabled. If the input is disabled, it becomes un-editable and un-clickable, although it might already contain a default value in it.
This is also a **boolean attribute**.

Eg. adding disabled attribute to an input element

```
<input type="text" name="country" value="India" disabled><br>
```

will make the input look like:



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

1.6. Checkboxes

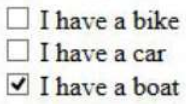
checkbox is a value of **type attribute** of input element. Check boxes are used when **more than one option may need to be checked** or you can also use it to enable or disable something.

The checkbox is shown as a square box that is ticked (checked) when activated. There is also a attribute named **checked**, that when present makes the checkbox selected by default when page loads.

Eg:

```
<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 below image:



1.7. Radio Button

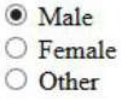
Radio button is just like a checkbox, but the difference is that the values of name attribute are all the same. To define a radio button the value of type attribute is "**radio**".

The name attributes are all set to the same value makes these radio buttons part of the same set, and therefore, you can **only select one of them at once**.

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:



1.8. fieldset and legend Elements

The **<fieldset>** element is used to provide **grouping** for a part of an HTML form. The <fieldset> tag draws a box around the related elements, which makes it more presentable.

The **<legend>** element is used for providing a title or explanatory **caption** for the rest of the contents of the legend element's parent element. <legend> comes just after the <fieldset> tag.

Eg, if a fieldset and legend is used

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

the form would look like this:

1.9. select Element

HTML **<select>** tag is used to create **drop down** list of options. Drop down list contains many options and the user can choose one of them.

The select tag also contains **name** attribute, like other form elements, that represent the associated data submitted to the server.

There are some of the unique attributes of select element-

- **multiple**, which specifies that multiple options can be selected.

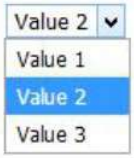
- **size**, which specifies how many options can be shown at once.

Eg:

```
<select name="select">
    <option value="value1">Value 1</option>
    <option value="value2" selected>Value 2</option>
    <option value="value3">Value 3</option>
</select>
```

Will show a dropdown like this:

and on clicking the above option will open other options as:



1.9.1. option Element

The tag used to define the possible options is **<option>** tag. This tag is put inside the <select> tag. For every option in drop down list, separate <option> element is used.

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

Each option element should have a **value** attribute, which contains the data value that will be submitted to the server when that option is selected.

1.9.2. optgroup Element

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

Eg:

```
<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:

EXTRA:

You can learn more about select and it's attributes from the link below:
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select

1.10. textarea Element

The **<textarea>** element is a input element where the user can input **multi-line text**, unlike <input> element where there is only a single line.

A text area can hold **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 like this:

1.10.1. rows and cols Attribute

These 2 attributes are used to set the size of <textarea>.
The **rows** attribute specifies the visible height of a text area.
The **cols** attribute specifies the visible width of a text area.

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

1.11. Submit Button

Submit button is a button that when clicked **automatically submits the form**. The button is defined at the end of the form. There are 2 different ways to add submit button to the form:

- via <input> tag
- via <button> tag

Both these ways will work in the same way.

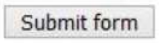
1.11.1. via input tag

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 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:



1.11.2. via button tag

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

1.11.3. 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.

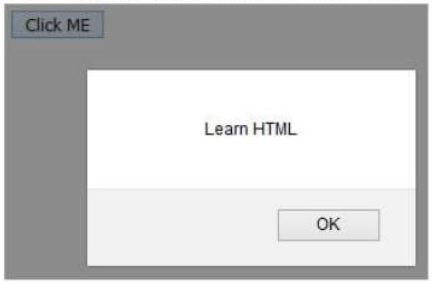
1.11.4. events on button

There are some events of HTML that we will cover here. Events are things that takes 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 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 like 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>
```

1.12. 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. To get the form to send the input data to the server, we need to set 2 attributes in the form:

- method attribute
- action attribute

Eg., the form has these attributes as:

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

1.12.1. method Attribute

The **method** attribute defines how the form data is send. The data can be send in different ways to the server. There is mainly 2 values we use to send the data:

- **get** - this appends the data into the url with '?' as separator in name-value pairs. Since this data will be visible, so sensitive data (like password) should not be send. 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 the sensitive data.

1.12.2. action Attribute

The **action** attribute defines where the form data is send when form is submitted. This contains the address (i.e. URL) of the file where the data is send. The URL can be provided in absolute and relative path.
The **absolute URL** points to another web site (like action="http://www.xyz.com/example.html").
The **relative URL** points to a file within the web site (like action="/example.html").

2. MEDIA ELEMENTS

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

2.1. Audio and Video Element

HTML can embed audios and videos directly in a web page without any external support. You can add audios and videos in a standard way.
The **<audio>** element is used to **add audio in web page**.
The **<video>** element is used to **add video in web page**.

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

The <audio> and <video> are same in the way the content is added to it and only tag name is different.

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 format is supported by the browser. Else, the text message will be shown.
The audio player will be shown like this:

2.1.1. 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.

2.1.2. source Tag

<source> element is used to **serve the same media content in multiple formats** so that different browsers can run any of the file that it supports. It is an empty element.

2.1.3. src Attribute

The **src** attribute is used to **specify the URL of the media file** that is needed to be played. This can have absolute or relative path.

2.1.4. 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**.

2.2. figure Element

The **<figure>** is a new tag introduces in HTML5. This element **specifies self-contained content, like illustrations,** diagrams, photos, code listings, etc.

Eg:

```
<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

2.2.1. figcaption Element

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

3. 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 favicon.

You can add 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:

4. META TAGS

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

There are 4 attributes that are used in meta tag:

- name
- content
- charset
- http-equiv

Meta tags have been one of the most basic elements of SEO. They are used to provide details about your site to search engines. **Search engine optimization** (**SEO**) is defined as the process of affecting the online visibility of a website or a web page in a web search engine's results.

Search engines such as Google often display the meta description in search results where they can **highly affect user visits to website**. So, it's very important to add meta tags to your web pages.

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

4.1. name Attribute

The **name** attribute is used to specify the name for the metadata. The name attribute is used together with content attribute. This attribute specifies a name for the information/value of the content attribute.
The name attribute can have one of the 6 values:

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

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.*

4.2. content Attribute

The **content** attribute gives the **value associated with the http-equiv or name attribute**.

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

4.3. charset Attribute

The **charset** attribute is used for **declaring the character encoding for the page**. It is a good practice to use ***UTF-8*** **encoding**. However, this must be taken care of that the declared character set matches the one on the page and is defined for every page of the website.
The syntax is: `<meta charset="character_set">`

4.4. 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 behavior.
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.