Remember:

* Document Type
* UTF
* Meta tags
* IFrame
* Entities
* URL Encoding
* XHTML
* Form
* HTML Apis
* Global Attribute
* URL Encoding
* HTTP message
* HTTP methods
* HTML accessibility
* Absolute and Relative

HTML (hypertext markup language)

It is a markup language used to define the structure of the document and how important components or content are which is already defined in the browser.

we can save HTML files using the extension .html(recommended) or .htm

HTML Tags are not case-sensitive

HTML Tags

Keywords between the angular bracket are called tags basically there are two tags opening and ending tag

HTML Elements

The combination of the opening tag and ending tag with some content between them is known as HTML Elements.

HTML Empty Elements or Tags

Some HTML elements have no content (like the <br> element). These elements are called empty elements. Empty elements do not have an end tag. But when we use react or other strict HTML versions then we need to use a self-closing tag i.e. <br/>

HTML Comments

The comment tag <!-- comment --> is used to insert comments in the source code. Comments are not rendered in the browsers.

You can use comments to explain your code, which can help you when you edit the source code at a later date. This is especially useful if you have a lot of code.

Text Formatting Tags

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <b> | Defines bold text |
| <em> | Defines emphasized text |
| <i> | Defines italic text |
| <small> | Defines smaller text |
| <strong> | Defines important text |
| <sub> | Defines subscripted text |
| <sup> | Defines superscripted text |
| <u> | Defines underlined text |
| <ins> | Defines inserted text by underlying the text |
| <del> | Defines deleted text by striking through the text |
| <s> | Defines text that is no longer correct, accurate or relevant by striking through it |
| <mark> | Defines marked/highlighted text |
| <pre> | Defines preformatted text, which is presented exactly as written in HTML |
| <tt> | Defines text appears as typed by a typewriter |
| <code> | Defines piece of computer code |
| <q> | Defines the short quoted text |
| <cite> | Defines reference to a cited work |
| <abbr> | Defines an abbreviation or acronym |

Important Tags and their attributes

IMAGE Tag

<img> tag is the self-closing tag which means that it doesn't contain the closing tag.

We can set the URL of the image in the src(source) attribute and if there is a problem with image loading or a network issue then we set the alt(alternative) attribute as the message.

There are also some attributes to control the size of the image i.e. width and height where the default measurement of the width and height is in pixels.

Example: <img src=” <https://www.linkedin.com/in/ashish-gupta003/> ” alt=” notes image” width=”100” height=”100” >

Anchor Tag

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

With the target attribute, you can define where the linked document will be opened. The target attribute has the following values:

* **\_self**: load the URL into the current tab itself. This is the default.
* **\_blank**: load the URL into a new tab or browser window.
* **\_parent**: load the URL into the parent browsing context. If there is no parent, this behaves the same as \_self.
* **\_top**: load the URL into the top-level browsing context. If there is no parent, this behaves the same as \_self.

Example: <a href=” <https://www.linkedin.com/in/ashish-gupta003/> ” target=” \_blank”> CLICK ME </a>

About Attributes

HTML Attributes can provide additional information about the HTML elements on your page and control their behavior.

**Some points to remember**:

* Attributes always come in name/value pairs like this: attribute\_name="value".
* Attributes are always added to the **start tag** of an HTML element.
* Attribute values should always be enclosed in quotes. Double style quotes (“ ”) are the most common, but single style quotes (‘ ’) are also allowed.
* In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes: name='John "ShotGun" Nelson' and vice-versa.

List

The list is grouped information about related entities which may be ordered or unordered

**Ordered Lists**

Ordered Lists is Numbered list and we can modify our ordered list using **type** and **start** attributes

Example:

Lists

1. first list item
2. secd list item
3. third list item

<h1> Lists </h1>

<ol type=”1” start=”1”>

<li> first list item </li>

<li> secd list item </li>

<li> third list item </li>

</ol>

**Unordered Lists**

An unordered List is an un-numbered list and we can modify our ordered list using **type** attributes i.e.

1. **type="disc"** - sets the list item marker to a bullet (default)
2. **type="circle"** - sets the list item marker to a circle
3. **type="square"** - sets the list item marker to a square
4. **type="none"** - the list items will not be marked

Example:

Lists

* first list item
* secd list item
* third list item

<h1> Lists </h1>

<ul type=” disc”>

<li> first list item </li>

<li> secd list item </li>

<li> third list item </li>

</ul>

Block and Inline Elements

**Block elements** are those that take up the **full width** available on a web page, effectively blocking out any other elements from sitting next to it on the **left or right**.

Example: <div>, <p>, <h1>, <nav>, etc.

**Inline elements** are those that only take up as much **width as needed** to display the contents of the element, thereby allowing other elements to be in line with the inline element.

Example: <b>, <span>, <img>, <a>, etc.

Class and ID

Classes and IDs are the global attribute which is used as the selector in CSS and JavaScript

The name of the Class and ID is **case sensitive**

HTML elements can have multiple classes separated by space but ID should be one

Class can be with multiple-element but ID should be unique for each Element.

Class is accessed using (.) and ID is accessed using (#) in CSS or JavaScript.

XHTML

XHTML stands for Extensible Hypertext Markup Language. It can be considered a part of the XML markup language this is because XHTML has features of both XML and HTML. XHTML is extended from XML and HTML. XHTML can be considered as a better version of HTML. In other words, XHTML is a stricter, more XML-based version of HTML.

**Differences from HTML**

* <!DOCTYPE> is mandatory
* The xmlns attribute in <html> tag is mandatory.
* <html>, <head>, <title>, and <body> are mandatory.
* Element must be properly nested.
* Element must be always closed.
* Element and attribute must be in lowercase.
* Attribute values must be quoted.
* Attribute minimization is forbidden.

**Syntax:**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">

<html xmlns=” <http://www.w3.org/1999/xhtml> “>

<head>

<title>Title of the Document</title>

</head>

<body>

<h1> hello XHTML </h1>

</body>

</html>

Tables

Tables are used to show the tabular data.

**Tags used in the table**

* <table> tag contains all the data in row-wise
* <caption> tag is used to give a caption to the table it is used only once.
* <thead>, <tbody> and <tfoot> tags are used to categories the content.
* <th> tag is used for column heading.
* <tr> tag is used to make rows in the table.
* <td> tag is used for inserting data (text, images, list, tables, etc.) in table rows.

**Attributes used in the table**

* Border attribute used to give border to the table data i.e., border=”1”
* rowspan attribute spans the row from left to right i.e., rowspan=”2”
* Colspan attribute spans the column from top to bottom. i.e., colspan=”3”

<table border=”1”>

<caption> Time Table </caption>

<thead>

<tr>

<th>Time</th>

<th>Room 1</th>

<th>Room 2</th>

<th>Room 3</th>

<th>Room 4</th>

</tr>

</thead>

<tbody>

<tr>

<td rowspan="3">9:00 AM - 10:00 AM</td>

<td rowspan="2">Keynote</td>

<td>Session A</td>

<td>Session B</td>

<td rowspan="3">Session C</td>

</tr>

<tr>

<td>Session D</td>

<td>Session E</td>

</tr>

<tr>

<td>10:30 AM - 11:30 AM</td>

<td colspan="2">Session F</td>

</tr>

</tbody>

<tfoot>

<tr>

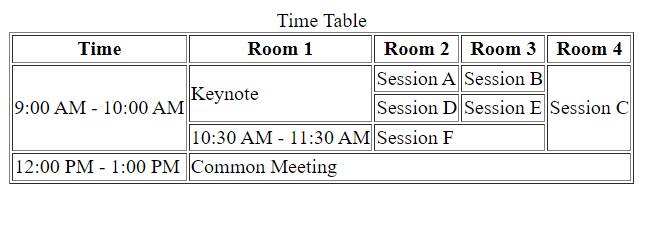
<td>12:00 PM - 1:00 PM</td>

<td colspan="4">Common Meeting</td>

</tr>

</tfoot>

</table>

OUTPUT:

Forms in HTML

HTML Forms are used to collect different kinds of input from the user. 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. it is defined with the <form>Top of the Form tag.

Example:

<form>

<-- form elements -->

</form>

Important attributes of form are:

* **Action Attribute:** The action attribute defines the action to be performed when the form is submitted. It tells 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 an absolute and relative path

**NOTE**: If the action attribute is not mentioned, the action is set to the current page URL.

* **Method Attribute**: The method attribute defines how the form data is sent. It specifies the HTTP method to be used when submitting the form data. The form data can be sent as URL variables (with method="GET") or as HTTP post transaction (with method="POST").

**NOTE**: The default HTTP method when submitting form data is GET.

**GET method:** This method appends the data into the URL with ‘?’ as a separator in name-value pairs. Since this data will be visible, so sensitive data (like passwords) should not be sent. This can be used to send query strings.



**POST method:** This method appends the data inside the body of the HTTP request. The post is used to send the sensitive data (the submitted form data is not shown in the URL).

**Syntax of action and method:**

<form action=” ./index.html” method=” POST”>

<-- form elements -->

</form>

* **Formaction Attribute:** The input formaction attribute specifies the URL of the file that will process the input when the form is submitted using the specific button.

**NOTE:** This attribute overrides the action attribute of the <form> element.

Example:

<form action=” ./user.html” method=” POST”>

<label for=” fname”> Full Name: </label>

<input type=”text” id=” fname” name=” fname”>

<input type=” submit” value=” **User**”>

<input type=” submit” value=” **Admin**” **formaction=” ./admin.html”**>

</form>

In the above example, if you submit by clicking the **“User”** button, the form will be submitted to **user.html** but submitting by **“Admin”** will submit the form to **admin.html**.

Form Elements and their attributes

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.

NOTE: It is an inline tag.

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

NOTE: type attribute is mandatory

Some more values for the type attribute are: -

* Submit
* Email
* Password
* Date
* Number
* Range
* URL
* Checkbox
* Radio
* Hidden
* Time

Value Attribute: Value is not a compulsory attribute to add to the input element. 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", and "image" - it defines the value associated with the input (It is also the value that is sent on submit)

name Attribute: The HTML form data is sent to the browser or server-side in the form of (name: value), where value is the entry, you provide respective to that named parameter. The name attribute is a compulsory attribute for input tags in a form. Without this attribute, this form element won't be submitted or, in other words, would not be sent to the server.

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

label Tag

A label tag describes the kind of input in a form, and it is not compulsory. You can do that without the use of a label tag. However, it is better to use the tag to describe the kind of input for the form element. NOTE: This is also an inline tag.

Example:

<form>

<label for=” fname”> First Name </label>

<input type=” text” id=” fname” name=” fname”>

</form>

**required Attribute:** It specifies that an input field must be filled out before submitting the form. Otherwise, it shows a pop-up to fill out the required field, or you can say it defines a mandatory field. The required attribute is a Boolean attribute.

**placeholder Attribute**: The placeholder attribute is used with the 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.

Select Tag

select tag is used to create a drop-down list of options. The dropdown list contains many options, and the user can choose one of them. The select tag also contains a name attribute, like other form elements, that represents the associated data submitted to the server.

<form>

<select name=” select”>

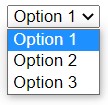
<option value=” originaloption1” selected> option 1 </option>

<option value=” originaloption2”> option 2 </option>

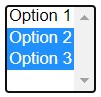
<option value=” originaloption3”> option 3 </option>

</select>

</form>



**Multiple Attribute**: It allows the user to select more than one value. (Press ctrl and select).



**Size Attribute**: specifies how many options can be shown at once.



Text Area and Fieldset

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

**rows and cols Attribute**

These 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.

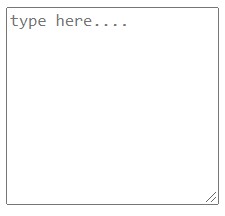
Example:

<form>

<textarea row=” 10” cols=” 20” placeholder=” type here…”></textarea>

</form>

Output:



# 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 provides 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.

Example:

<form>

<fieldset>

<legend>DETAILS:</legend>

<label for=” fname”> Name: </label>

<input type=” text” id=” fname” name=” fname”><br>

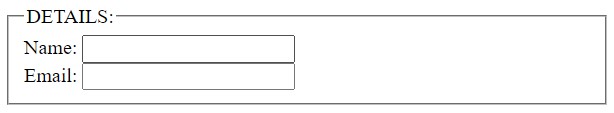
<label for=” email”> Email: </label>

<input type=” email” id=” email” name=” e-mail”>

</fieldset>

</form>

Output:



Type Submit:

The submit button is a button that, when clicked, automatically submits the form. The button is defined at the end of the form. There are two different ways to add a submit button to the form, and both of these ways will work in the same way i.e., via <input> tag or <button> tag.

submit as input vs submit as button

The <input> tag can also be used to create a button. To use it as a button, the type attribute is set to 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 <button> tag is also used to create a submit button in the form. Although, the <button> tag also creates a submit button, but there are some benefits of a 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 to the button. The button has to set the type attribute to submit to make it a submit button.

Interview Questions



**Q1. What is HTML form validation?**

**Ans 1.** HTML form validation is a process of examining the HTML form page’s contents to avoid errored-out data being sent to the server. This process is a significant step in developing HTML-based web applications, as it can easily improve the quality of the web page or the web application. There are two ways to perform the HTML form Validation, and they are by Using HTML5 built-in functionality and by Using JavaScript.

**Q2. What is the usage of a novalidate attribute for the form tag that is introduced in HTML5?**

**Ans 2.** Its value is a Boolean type that indicates whether or not the data being submitted by the form will be validated beforehand. By making this false, forms can be submitted without validation which helps users to resume later also.

|  |
| --- |
| <form action = "" method = "get" **novalidate**>  Name:<br><input type="name" name="sname"><br>  Doubt:<br><input type="number" name="doubt"><br>  <input type="submit" value="Submit"> </form> |

**Q3. What are the different new form element types in HTML 5?**

**Ans 3.** Following is a list of 10 frequently used new elements in HTML 5:

|  |  |
| --- | --- |
| * Color * Date * Datetime-local * Email * Range * Telephone | * Time * Url * Number * Search |



1

**Q4. What is a button tag?**

**Ans 4 .** The button tag is used in HTML 5. It is used to create a clickable button within the HTML form on the web page. It is generally used to create a "submit" or "reset" button. Let's see the code to display the button.

<button name="button" type="button">Click Here</button>

**Q5. What is the use of the required attribute in HTML5?**

**Ans 5 .** It forces a user to fill in text on the text field or text area before submitting the form. It is used for form validation.