HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

- All HTML tags must be enclosed within < > these brackets.
- Every tag in HTML performs different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

Syntax

<tag> content </tag>

HTML Tag Examples

Note: HTML Tags are always written in lowercase letters. The basic HTML tags are given below:

 Paragraph Tag

<h2> Heading Tag </h2>

Bold Tag

Italic Tag

<u> Underline Tag</u>

Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

**
 Tag**: br stands for break line, it breaks the line of the code.

<hr> Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

HTML Meta Tags

DOCTYPE, title, link, meta and style

HTML Text Tags

, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, , , <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and

HTML Link Tags

<a> and <base>

HTML Image and Object Tags

<img>, <area>, <map>, <param> and <object>

HTML List Tags

, <ol>, <li>, <dl>, <dt> and <dd>

HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

HTML Form Tags

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

HTML Scripting Tags

script and noscript

Note: We will see examples using these tags in later charters.

All HTML Tags

Let's discuss the complete tag list that is present in the HTML that is represented below.

1. Comments:

With the help of this tag, we can add comments in the HTML. After commenting on the elements, the browser does not render it. Also, these comments parts are not displayed on the web page. But these commented codes are available in the source code.

Syntax:

<strong**><**!-- --**></**strong**>**

2. Doctype:

With the help of this tag, we can declare the version of the markup language to the web browser.

Syntax:

< !DOCTYPE html **>**

3. <a>: Anchor

With the help of this tag, we can create a hyperlink to the web page. The hyperlink can be done with the help of another file in a different location.

Syntax:

<a href="url"**>**Link Text**</**a**>**

4. <abbr>:

With the help of this tag, we can define the abbreviations or acronyms of longer words, such as www, HTML, HTTP, etc.

Syntax:

```
<abbr title="expanded-text">abbreviation</abbr>
```

5. <address>:

With the help of this tag, we can find the address or the contact details of the web page's author. The <address> tag contains addresses, cellphone numbers, emails, website connections, etc.

Syntax:

```
<address>
Contact Information Goes Here
</address>
```

6. <area>:

With the help of this tag, we can identify the clickable area associated with the hyperlink.

Syntax:

```
<area shape="shape-type" coords="coordinates" href="destination-url" alt="alternative-text">
```

7. <article>:

With the help of this tag, we can represent a piece of content that is used to be distributed to other websites without being a part of this. With the help of <article> tag, we can display entire blog posts, news articles, or similar content, etc.

Syntax:

```
<article>
<!-- Content goes here -->
</article>
```

8. <audio>:

With the help of this tag, we can embed the audio file into the webpage. To do this, there is no need for additional plug-ins like Flash.

Syntax:

```
<audio src="audio-file.mp3" controls>
Your browser does not support the audio element.
</audio>
```

9. <aside>:

With the help of this tag, we can define the content related to the web page's main content, but they are not constituted to the web page.

Syntax:

```
<aside>
<!-- Content goes here -->
</aside>
```

10. <b>:

With the help of this tag, we can make the text bold without any additional significance.

Syntax:

```
<b>Text to be displayed in bold</b>
```

11. <base>:

With the help of this tag, we can create the target address, which is targeted by all the URLs present in the document.

Syntax:

```
<base href="base-url" target="base-target">
```

12. <bdi>:

With the help of this tag, we can tell the browser to isolate the text that comes from other destinations with the help of a bidirectional algorithm.

Syntax:

```
<bdi>Text with different directionality</bdi>
```

13. <bdo>:

With the help of this tag, we can override the current text.

Syntax:

```
<bdo dir="direction">Text with overridden directionality</bdo>
```

14. <blockquote>:

With the help of this tag, we can provide a way to insert the quotation in the document from another source. By default, the browser takes the text inside a quote tag as indented.

Syntax:

```
<blockquote>
Quoted content goes here.
</blockquote>
```

15. <body>:

With the help of this tag, we can define the web page's main content. The body tag includes text, paragraphs, headings, images, tables, links, and videos.

Syntax:

```
<body>
  <!-- Content goes here -->
</body>
```

**16.
:**

With the help of this tag, we can produce a line break on the web page. We can use this tag basically during the writing of poems or addresses that require line breaks.

Syntax:

17. <button>:

With the help of this tag, we can create a clickable button on the web page. We can use the button tag in forms or anywhere in a document.

Syntax:

<button>Button Text</button>

18. <canvas>:

With the help of this tag, we can define a particular region used for drawing graphics through scripting (usually JavaScript).

Syntax:

1. <canvas id="canvas-id" width="width" height="height"></canvas>

19. <caption>:

With the help of this tag, we can create the table caption, which is used to create the title for the table.

Syntax:

<table>

<caption>Caption Text</caption>

<!-- Table content goes here -->

</table>

20. <cite>:

With the help of this tag, we can do some creative work and give a title for that creative work.

Syntax:

<cite>Referenced Content</cite>

21. <code>:

With the help of this tag, we can insert a particular code fragment, which might be HTML or CSS code.

Syntax:

<code>Code goes here</code>

22. <col>:

With the help of this tag, we can specify all the column's properties. We can specify it inside the <colgroup> element.

Syntax:

```
<col>
```

23. <colgroup>:

With the help of this tag, we can specify one or more columns in a group. With the help of this, we can apply the style to the entire column without repeating the style for every column and every row.

Syntax:

```
<colgroup>
<!-- col elements go here -->
</colgroup>
```

24. <data>:

With the help of this tag, we can add machine-readable translation. The processor generates it, and the browser can generate the human-readable value.

Syntax:

```
<data value="data-value">Data Label</data>
```

25. <datalist>:

With the help of this tag, we can provide an auto-complete feature to the form element. With the help of this, the user can select the option from the option list.

Syntax:

```
<input list="datalist-id">
<datalist id="datalist-id">
<!-- option elements go here -->
</datalist>
```

26. <dd>:

With the help of this tag, we can define the definition description in the HTML. It is also used to describe an item in the list of descriptions.

Syntax:

```
<dl>
<dt>Term</dt>
<dd>Description or definition goes here</dd>
</dl>
```

27. <del>:

With the help of this tag, we can declare the particular section that will be deleted from the document. After doing this, the web browser strike over that particular section in the webpage.

Syntax:

```
<del>Deleted Text</del>
```

28. <details>:

With the help of this tag, we can create a toggle. This toggle contains all the information but is in a closed state. When we mouse hover over it, it comes to an open state and displays all the hidden information.

Syntax:

```
<details>
<summary>Summary Text</summary>
<!-- Content goes here -->
</details>
```

29. <dfn>:

With the help of this tag, we can represent the definition element. With the help of the definition element, we can define the instances of the definition. When a term first appears in the document, it is known as the instances of the definition.

Syntax:

```
<dfn>Term Definition</dfn>
```

30. <dialog>:

With the help of this tag, we can define a dialogue box and the other webpage components. With the help of this component, the webpage can interact with the user beautifully.

Syntax:

```
<dialog open>
<!-- Dialog content goes here -->
</dialog>
```

31. :

With the help of this tag, we can create a generic container that can hold the flow of the content. It also does not have the default rendering.

Syntax:

```
<div>
<!-- Content goes here -->
</div>
```

32. <dl>:

With the help of this tag, we can define the list of items on the web page. We can also represent the metadata with the help of this tag.

Syntax:

```
<dl>
```

```
<dt>Term 1</dt>
<dd>Definition 1</dd>
<dt>Term 2</dt>
<dd>Definition 2</dd>
<!-- more terms and definitions -->
</dl>
```

33. <dt>:

With the help of this tag, we can define the definition list on the web page. we have to write the <dt> tag within the <dl> tag. We can also flow this element with the help of <dd> tag.

Syntax:

```
<dl>
<dt>Term</dt>
<!-- Definition or description goes here -->
</dl>
```

34. :

With the help of this tag, we can emphasize a particular word or phrase within a sentence. It converts the normal word or sentence into italic and semantic styles.

Syntax:

```
<em>Emphasized Text</em>
```

35. <embed>:

With the help of this tag, we can embed external things such as audio or video into the webpage.

Syntax:

```
<embed src="content-url">
```

36. <fieldset>:

With the help of this tag, we can create a group of related elements inside a form. Also, with the help of this, we can create a box around the element.

Syntax:

```
<fieldset>
<legend>Fieldset Title</legend>
<!-- Form controls go here -->
</fieldset>
```

37. <figcaption>:

With the help of this tag, we can attach the caption for the image, table, or chart contained within the <figure> element.

Syntax:

```
<figure>
<!-- Content goes here -->
<figcaption>Caption or Description</figcaption>
</figure>
```

38. <figure>:

With the help of this tag, we can create self-contained content, also referred to as a single entity.

Syntax:

```
<figure>
<!-- Content goes here -->
<figcaption>Caption or Description</figcaption>
</figure>
```

39. <footer>:

With the help of this element, we can represent the footer part in the root element.

Syntax:

```
<footer>
<!-- Footer content goes here -->
</footer>
```

40. <form>:

With the help of this tag, we can receive the information from the user.

Syntax:

```
<form action="form-handler-url" method="HTTP-method">
<!-- Form controls go here -->
</form>
```

41. <head>:

With the help of this tag, we can specify the header part of the header section of the code. It contains all the information regarding the document.

Syntax:

```
<head>
  <!-- Head content goes here -->
</head>
```

42. <header>:

With the help of this tag, we can define the webpage's header. It also contains information regarding the title and heading of the document and section.

Syntax:

```
<header>
  <!-- Header content goes here -->
</header>
```

43. <hgroup>:

With the help of this tag, we can represent the heading group. It is also used for group heading elements, such as headings and subheadings, from <h1> to <h6>.

Syntax:

```
<h1>Main Heading</h1>
<h2>Subheading 1</h2>
<h3>Sub-subheading 1.1</h3>
<h3>Sub-subheading 1.2</h3>
<h2>Subheading 2</h2>
```

44. <h1> to <h6>:

With the help of this tag, we can define the web page's heading. It also makes the heading as large and bold by default. the <h1> headings are displayed in the largest font, whereas the <h6> headings are displayed in the smallest font.

Syntax:

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
```

45. <hr>:

With the help of this tag, we can divide the web page. It stands for horizontal rule or theme break.

Syntax:

```
<hr>
```

46. <html>:

With the help of this tag, all other elements are contained by it. It is also known as the root of the HTML element.

Syntax:

```
<!DOCTYPE html>
<html>
```

<!-- HTML content goes here -->

</html>

47. :

With the help of this tag, we can make any content italicized. It Highlights an important term, phrase, or technical term.

Syntax:

Italicized Text

48. <iframe>:

With the help of this tag, we can embed another HTML page into the current webpage.

Syntax:

<iframe src="url"></iframe>

49. <img>:

With the help of this tag, we can embed an image into the current web page. It links an image to the image without directly inserting it into the web page.

Syntax:

<img src="image-url" alt="alternative-text">

50. <input>

With the help of this tag, we can enable interactive control in the form. It also depends on the various fields of the attribute type.

Syntax:

<input type="input-type" name="input-name" value="default-value">

51. <ins>:

With the help of this tag, we can specify the block of the inserted text. It also signifies that a particular range of text has been added to the document.

Syntax:

<ins>Inserted Text</ins>

52. <kbd>:

With the help of this tag, we can refer to the textual part of the inline text that represents the user's keyboard, voice, or any other text input device.

Syntax:

<kbd>Keyboard Input</kbd>

53. <label>:

With the help of this tag, we can improve the usability for mouse users, i.e., clicking on the label will toggle the control.

Syntax:

```
<label for="input-id">Label Text</label>
```

54. <legend>:

With the help of this tag, we can define the title for the child's content. Legends are the parent element of all the child elements. We can define these elements with the help of <fieldset>.

Syntax:

```
<fieldset>
<legend>Fieldset Caption</legend>
<!-- Form controls go here -->
</fieldset>
```

55. :

With the help of this tag, we can represent the list of items on the web page. It contains an ordered list (an unordered or a menu) as a parent element.

Syntax:

```
<ul>
<li>List Item 1</li>
  <li>List Item 2</li>
<li>List Item 3</li>
</ul>
```

56. <link>:

With the help of this tag, we can link external resources to the current webpage.

Syntax:

```
<link rel="stylesheet" href="stylesheet.css">
```

57. <main>:

With the help of this tag, we can make an ordinary element the most important element. We have to write this tag inside the <body> tag.

Syntax:

```
<main>
<!-- Main content goes here -->
</main>
```

58. <map>:

With the help of this tag, we can define the map image. The image map also has a clickable area.

Syntax:

```
<map name="map-name">
```

<!-- Area tags go here -->

</map>

59. <mark>:

With the help of this tag, we can represent the text as highlighted, which will be used for reference or notation purposes. Based on the importance of the content, it highlighted the text.

Syntax:

<mark>Highlighted Text</mark>

60. <meta>:

With the help of this tag, we can represent the document's metadata. The metadata contains page description, keywords, copyright, the language used, and the author.

Syntax:

<meta name="metadata-name" content="metadata-value">

61. <meter>:

With the help of this tag, we can represent the Measurements within a given range or fractional values. It is also known as the gauge.

Syntax:

<meter value="current-value" min="minimum-value" max="maximum-value">Fallback Content</meter>

62. <nav>:

With the help of this tag, we can represent the section of navigational links in a document. Some examples of nav tags are Menus, tables of contents, indexes, etc.

Syntax:

<nav>

<!-- Navigation links or menus go here -->

</nav>

63. <noscript>:

With the help of this, we can provide an alternative script when the user's browser does not support the particular script. It also works whenever the user's browser does not support client-side scripting or does not support any scripting language.

Syntax:

<noscript>

<!-- Alternate content goes here -->

</noscript>

64. <object>:

With the help of this tag, we can display multimedia like audio, videos, images, PDFs, and Flash on web pages.

Syntax:

```
<object data="resource-url"></object>
```

65. :

This tag is used for the creation of the ordered list. The ordered list can be numerical or alphabetical. Inside the <ol> tag, we can create the list using the <li> tag. It will then follow the order.

Syntax:

```
<ol>
<li>First Item</li>
<li>Second Item</li>
<li>Third Item</li>
</ol>
```

66. <optgroup>:

This tag is used in the creation of the drop-down list. With the help of this tag, we can define the group of options.

Syntax:

```
<select>
<optgroup label="Group Label">
   <option value="option1">Option 1</option>
<option value="option2">Option 2</option>
<!-- Additional options -->
</optgroup>
<!-- Additional outgroup or option elements -->
</select>
```

67. <option>:

With the help of this tag, we can define the option in the drop-down list by using the <select> tag.

Syntax:

```
<select>
<option value="option-value">Option Text</option>
<!-- Additional options -->
</select>
```

68. <output>:

With the help of this tag, we can create the container for the element that injects the result of the calculation.

Syntax:

```
<output for="input-id"></output>
```

69. <param>:

With the help of this tag, we can define the parameter for an object. With the help of this, we can pass the parameter for another embedded object.

Syntax:

```
<object>
<param name="param-name" value="param-value">
</object>
```

70. <picture>:

With the help of this tag, we can provide flexibility to the web page by specifying the image. It also contains tags like <source> and <img>.

Syntax:

```
<picture>
<source srcset="image1.jpg" media="media-query">
<source srcset="image2.jpg" media="media-query">
<img src="fallback-image.jpg" alt="Fallback Image">
</picture>
```

71. <pre>:

With the help of this tag, we can declare the block of pre-formatted text on the web page the pre-formatted character, such as spaces, line breaks, tabs, etc.

Syntax:

```
<pre>
The pre-formatted text goes here.
</pre>
```

72. <progress>:

With the help of this tag, we can display the progression of the task. With the help of this, the developer can create a progress bar on their web page.

Syntax:

```
<progress value="current-value" max="maximum-value"></progress>
```

73. <q>:

With the help of this tag, we can create an inline quotation on the web page. The text is contained inside the quotation marks in most browsers today.

Syntax:

```
Here is a <q>short quotation</q> within a sentence.
```

74. <s>:

With the help of this tag, we can render the text with a strikethrough or a line through it. We can use this tag when Things are no longer accurate or relevant.

Syntax:

```
<s>Struck-through text</s>
```

75. <samp>:

With the help of this tag, we can define the sample output of the computer program. The content of this tag is displayed using a monospaced font in the browser.

Syntax:

```
<samp>Sample text or code goes here</samp>
```

76. <script>:

With the help of this tag, we can insert the client-side scripting. It also enables the insertion of the script in the HTML web page.

Syntax:

```
<script src="script.js"></script>
```

77. <section>:

With the help of this tag, we can create the header and footer or any other section on the web page.

Syntax:

```
<section>
  <!-- Content goes here -->
</section>
```

78. <select>:

With the help of this tag, we can define the list of options inside the list.

Syntax:

```
<select>
  <option value="option-value">Option Text</option>
  <!-- Additional options -->
</select>
```

79. <small>:

With the help of this tag, we can reduce the font size by one size smaller than its base size, for instance, from large to small.

Syntax:

```
<small>Small text goes here</small>
```

80. <source>

With the help of this tag, we can link multiple media resources like <audio> and <video>,

Syntax:

```
<source src="media-file.mp4" type="video/mp4">
```

**81. **

With the help of this tag, we can define the content as very important. The browser will display that element in bold.

Syntax:

```
<strong>Emphasized text</strong>
```

82. :

It is the most generic element in the HTML. We can apply the style by grouping the elements.

Syntax:

```
<span>Text or content goes here</span>
```

83. <style>:

We can provide the style information for the document with the help of this tag.

Syntax:

```
<style>
  CSS rules go here
</style>
```

84. <sub>:

With the help of this tag, we can display the subscript element only for typographical reasons.

Syntax:

```
<sub>Subscript text</sub>
```

85. <summary>:

With the help of this tag, we can create a box with the help of <details> element that contains the summary, caption, or legend. When we click on the summary button, it performs the open/closed state of the parent *details> element.

Syntax:

```
<details>
<summary>Summary text</summary>
<!-- Collapsible content goes here -->
</details>
```

86. <sup>:

With the help of this tag, we can display any text as the superscript of the inline element.

Syntax:

```
<sup>Superscript text</sup>
```

87. <svg>:

With the help of this tag, we can embed the SVG graphs into the web page. Also, we can define the vector graphics in an XML format with the help of this tag.

Syntax:

```
<svg>
  <!-- SVG content goes here -->
</svg>
```

88. <table>:

With the help of this tag, we can define the data in a tabular format in HTML. It will create a two-dimensional table that has some rows and columns.

Syntax:

```
<table>
  <!-- Table content goes here -->
</table>
```

89. <tbody>:

With the help of this tag, we can encapsulate the table's rows associated with the table body.

Syntax:

```
<table>
  <tbody>
    <!-- Table rows go here -->
  </tbody>
</table>
```

90. <td>:

With the help of this tag, we can define the table's cell. A table row may consist of more than one <td> element. The default alignment is left in this element.

Syntax:

```
<tr>
  <td>Table cell content</td>
</tr>
```