1. Introduction to html

Hypertext markup language. (Tim Berners lee). It is the basic building block of the web.

Hyper -> to move one page to other page using hyper link.

Markup -> how text is processed and presented. language defines the text document within the tag that define the structure of web pages

<head>, <title>,<body>,<div>

1. Html Element structure, content Tags, nesting Element and Empty Element

Tag Tag

<p> content </p>

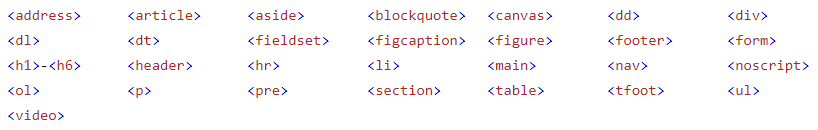
-------------------------

Html Element

Empty Element -> inline block

1. Block and inline elements

Block -> block level elements makes visible block on a page. Takes a new line. (structural element). It will not place in inline element.



Inline -> this were surround only small part (width) document content and it can be placed in block element. Will not take new line.

A picture containing text, screenshot, font, line

Description automatically generated

Void Elements:-

There is a special group of elements that only have start tags and does not contain any content within it, these elements are called void elements. Void elements doesn’t have ending tags and can only have attributes but do not contain any kind of content. These elements can have backslash before ending of start tag but that is completely optional. Example of such elements are <br>, <hr>, <img>, <input>, <link>, <base>, <meta>, <param>, <area>, <embed>, <col>, <track>, <source> etc.

1. HTML Attributes.

<p class=”some-class”> content</p>

-----------------------

Attributes

Extra information about element that never appear in the content.

Boolean attributes :- don’t have value (disable).

1. Html document structure  
      
   <!DOCTYPE html> - were meant to act as link to a set of rules that html page had to follow to be considered good html.

<html lang=”en-US”> -> root element.

<head> -> act as container. It will not appear in view.

<title>page1<title>

<head>

<body>

<p>hello world</p>

</body>

</html>

<meta charset=” utf-8”> => human written language it supported. It can handle all languages.

<title> => it will show in our bookmarks, for use in search results.

1. Entities for special characters in content

Single quotes ->

Double quotes ->

If we are using same quotes in inside also, we need to use entities.

&apos; -> ‘

whitespace -> no matter how much whitespace we use inside html content and line break.

and convert single break in view.

Special characters -> <,>”,’ to &lt, &gt, &quot , &apos

1. Metadata in html.

The data which describes the html document.   
<meta charset=” utf-8”> => human written language it supported. It can handle all languages.

<meta name=” author” content=”test123”>

Name what is title

Content tells description.

<meta name=”twitter: title” content=”Mozilla Developer Network”>

Favicon -> 16px png or gif.

Css <link href=”” />

Js <script src=”” />

<span lang=” ja”> </span>

1. Html text formatting.  
     
   heading <h1 to h6> and paragraphs <p>.

One single h1 in a page we need to use.

In 6 headings we need to no more than 3 per page.

h1 -> search engines and screen readers advantage.

Html list :-

Unorder list -> doesn’t matter on items order. <ul> black dot will come.

Order list -> which the order of the items does matter. <ol> number will come default.

ol start="2"

then number start with 2,3,4 in list instead of 1.

Emphasis:-

<em> -> certain words important for that sentence or different in some way. Particular word we need to speak with stress or different tone voice. It styles as italic font.

Strong :-

It tells the word is more importance in sentence.

We can nested tag <strong> and <em>

Italic:- <i>

Bold:- <b>

Underline:- <u>

This will affect presentation and not semantics. Presentational elements.

Create HyperLinks: -

Allow us to link documents to other documents. Href (hypertext reference).

<a href=” <https://www.google.com>” title=” google”>home</a>

Using the title, it will show data in tooltip hovering.

URL – uniform resource locator.

Absolute path: - always point same location no matter where its used. Ex- home.html.

Relative path: - relative to this file. Ex – projects/home.html or ../home.html.

<a href=” #test”> test</a> to navigate to particular location of page.

<h2 id=”test”>test</h2>

Description list in html. dl, dt, dd elements: -

Dl -> description list.

Dt -> description term.

Dd -> description definition.

<dl>

<dt>hello</dt>

<dd>hello world<dd>

</dl>

Block Quote and q element: -

“Block quote” Marking up quotations, it shows content text in block level. It adds padding and margin to content. It has “cite” attribute.

‘q’ show content text in inline level. “” it will add. It has attribute cite.

Represent programming code:-

Code: - if you have some small programing code we need to show in in a style fashion. f

Pre: - for retaining white space. It show with white space .

Kbd:- keyboard input element like Ctrl+s.

Var:- marking up variable name. some special style add.

Samp:- represents sample output from a computer program.

1. What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

* Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.
* Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.

1. Images in HTML

Attributes (src, alt, title, height, width).

Annotating images: -

<figcaption> -> to add caption for image. (we can use video, audio, snippets).

<figure>

<img src=””/>

<figcaption> title<figcaption>

</figure>

1. Video and audio

In old days we use flash. From html5 audio , canvas, iframe and video came place.

Video allows attributes height, width and audio not supported.

Audio not supported poster.

<video controls autoplay loop muted preload=” auto” poster=”.png”>

<source src=”.mp4” type=” video/mp4”>

<source src=”. webm” type=” video/webm”>

<p> your browser not supported html5</p>

</video>

<audio controls autoplay loop muted preload=” auto” >

<source src=”.mp3” type=” audio/mp3”>

<source src=”. ogg” type=” audio/ogg”>

<p> your browser not supported html5</p>

</audio>

1. Object to Iframe

Embedding other web pages and other than pdf, svg, flash.

In olden days we use <object>, <embed> it has problems security and file size.

Border: none. Not allow border.

Sandox- request heightened security setting.

frameborder=”0”

referrerpolicy=”no-referrer-when-downgrade”

<iframe src=”” width=”” height="” allowfullscreen sandox>

<p>fallback link for browser iframe not supported</p>

</iframe>

1. Adding vectors graphics to the web

Small file size and highly scalable so they don’t pixelate when zoomed in or blown up to large size. (Svg image cached by browser and loading faster) don’t have fall back.

Raster images: - png, jpg, gif, bmp(bitmap). When zoom blur will come.

Vector images: - shape and path direction (svg). Lighter than raster images.

<svg version=”1.1” width=”30” height=”50” xmlns=” https://w3.org/2000/svg”>

<rect width=”100%” height=”100%” fill=” black”/> <circle cx=”10” cy=”20” fill=” red” r=”20”/>

</svg>

<image src=”.svg” alt=”svg”/>

1. Responsive image

<img srcset=” small.jpg 500w, mediam.jpg 800w, large.jpg 1200w” />

(or)

Sizes=”(min-width:1300px) 600px” in above tag only.

1. Simple html table   
     
   <table>

<tr> -> table row

<td>usa</td> -> table data

<td>uk</td>

<td>india</td>

<tr>

</table>

<table>

<tr>  
    <th>Company</th>  
    <th>Contact</th>  
    <th>Country</th>  
  </tr>

<tr>

<td>person1</td>

<td>24</td>

<td>usa</td>

</tr>

<tr>

<td>person2</td>

<td>24</td>

<td>usa</td>

</tr>

</table>

Cell spacing: - used to remove space b/w cells in table tag we need to use.

rowSpan: - for two rows we need to combine in ‘td’ we need to use rowspan.

colSpan: - for two columns we need to use same heading. In ‘th’ we need to use colspan.

colGroup: - we need to use add color or bg-color on table. Particular cell or column. Bellow table we need to use here we need to use span=”2” to share styles.

Caption element:- inside table tag we need to use caption tag.

Thead:- organize table in good form.

Tbody:- organize table in good form.

Tfoot:- organize table in good form.

1. Web forms  
     
   main point of interaction b/w a user and web site or application. Web form contain one or more form controls. Nested form does not allow.

<form action=” url” method=” post”>

</form>

Fieldset: - which can group all form controls. Attributes “disable”, “name”.

Legend: - to create the form type it came like label with border.

Label: - meaning for the input box. for=”test” and id=”name”.

For input type=” hidden” it will not appear in ui.

readonly: - user no need to write data not able to edit.

disabled: - disabled in grey color. Not able to edit. We are not able to send data to the server.

Multiple: - which allows us to multiple values. Email, select.

Search: - it will give close icon to clear text.

Tel: - for phone numbers. In mobile only the number keypad open.

url: - it allows well-formed url.

Number: - min, max, step.

Slider: - default (0 to 100), min, max, value.

Date and time: - type=” datetime-local”.

Textarea: - cols (width), rows (height), wrap.

Progress (tag): - max, value.

Meter (tag): - min, max, value, low, high.

Lower part: - b/w min and low.

Medium part: - b/w low and high.

Higher part: - b/w max and high.

Pattern: - “[0-9]{10}”, “.+gmail.com”.

Enctype=” multipart/form-data”.

type=” file”. We need to use multipart/form-data.

Novalidate:- no need to use html default validation.

**17.** **Define multipart form data?**

Multipart form data is one of the values of the enctype attribute. It is used to send the file data to the server-side for processing. The other valid values of the enctype attribute are text/plain and application/x-www-form-urlencoded.

1. How to optimize website assets loading?

To optimize website load time we need to optimize its asset loading and for that:

CDN hosting - A CDN or content delivery network is geographically distributed servers to help reduce latency.

File compression - This is a method that helps to reduce the size of an asset to reduce the data transfer

File concatenation - This reduces the number of HTTP calls

Minify scripts - This reduces the overall file size of js and CSS files

Parallel downloads - Hosting assets in multiple subdomains can help to bypass the download limit of 6 assets per domain of all modern browsers. This can be configured but most general users never modify these settings.

Lazy Loading - Instead of loading all the assets at once, the non-critical assets can be loaded on a need basis