

HTML 5

HyperText Markup language [HTML] is a text markup language used on the World Wide Web. It tells the browser how to display a web page. Elements in HTML consist of alphanumeric tags within angle brackets.

Basic HTML TAGS.

```

<!DOCTYPE html> // this is a HTML 5 document
<html lang="en"> // language = english.
<head> // starting head tag which contains
supplementary information about the document.
</head> // closing head tag.
<body> // starting of body tag which contains the
actual information of the page and appropriate
markup tags that are needed to render the page.
<br> // single tag which displays a horizontal line
<!-- comment text --> html style comment.
<TITLE> the title of the page <TITLE> // used inside
the head tag to specify the title of the page.
<H1> Largest and most important heading </H1>
<P> Paragraph tag </P> // used inside body tag.
<B> Bold's the text </B> <U> Underlines </U>
<I> Italic text </I> <big> big text </big>.
<H6> last or least important heading </H6>.
<br> single tag for line breaks.
<DIR> directory listing tag </DIR>
<STRIKE> text will have a horizontal cut from centre </STRIKE>
<MARK> highlighted text </mark>.

```

- Nesting of HTML tags.

HTML tags should be nested such that they surround HTML tags. HTML tags should not crossover. So we nest them as. `<I> the text <I>`.

- Linking a webpage.

` link text <a>`

- Attribute of anchor tag.
- Starting anchor tag.

- Linking a folder on the webpage.

` link text `

a folder or directory stored with .html extension.

- Linking an image on the webpage.

``

- Subscript & Superscript.

`<P> C0 ₂ </P> // C02`

// used for writing mathematical expressions.

- Code text.

`<pre> these code lines or data must be displayed as it is with some space and format </pre>`

- Input date.

`<input type = " datetime-local " />` defines a date & time picker.

- Drop downs.

`<select> and <datalist>` elements help create a drop down menu. "Multiple" attribute can be used for multi select drop down. `<option>` and `</option>` creates value attribute uses for multiple instances.

Basic HTML Format.

```

<!DOCTYPE html> // this document is html type.
<html lang="en"> // html language english.
<html> // start of html document.
<head> // head or top part of the document.
<title> // title of the webpage. </title>
<meta> // contains data about data.
<base> // specifies url and target for all links on a page
<style> // adds style information on page </style>
<link> // links to an external style sheet
<script> // contains CSS and JS information </script>
</head> // end tag for heading tag.
<body> // contains actual contents of the page
<header> // introductory content & navigational aids.
</header> // end tag of the header tag.
<main> // central data of the document and how
it should be presented. </main>
<footer> // contains authority or copyright information
it contains the bottom most element of page </footer>
</body> // End tag of body tag.
</html> // End of html document.

```

<header>

<h1>

<main>

<section>

<article>

<aside>

<footer>

- HTML TAGS FOR Page Manipulation:
 - Break
 // provides page break as many time is mentioned, no end tag required.
 - Division <DIV> // unique section or division, useful for conjunction with style sheet </DIV>
 - Block Quotes <BLOCKQUOTE> // enclosed larger block
 - Lists // lists are of three basic types.
 - ordered lists // numbering scheme.
 - unordered lists // bulleting scheme.
 - <DL> definition lists. </DL> // glossary scheme.
 - Address <ADDRESS> // a block element tag with format set for writing address on the page </ADDRESS>
 - Tables <TABLE> // creates a table in which an optional caption can be specified with <caption> text </caption> The <THEAD> can specify a header section then a footer section can be specified <TFOOT> and Body section with <TBODY>. Rows are specified with <TR> table row </TR> <TH> represents headings <TH> and <TD> table data or content <TD> <COL> for columns </COL> <COLGROUP> for a group of columns </COLGROUP> </TABLE>
 - Fonts // a font element lets you include a font element or style sheet rule that sets fonts <Basefont> is used for styling the whole document </Basefont>
 - Calculation: <form oninput> takes the calculative action and displays output with other tag

Syntax: <form oninput = "result.value = parseInt(a.value) + parseInt(b.value)">

Enter inputs with <input.....> and <output></output></form>

• HTML Attributes

ALIGNMENT // used in heading and paragraph tags.

<P ALIGN = "CENTER"> the text here is center aligned </P>

<P ALIGN = "RIGHT"> the text aligns to right </P>

<P ALIGN = "LEFT"> text left aligned </P>.

LISTING // used in making ordered/unordered &

definition list using, numerals, bullets or glossary.

<OL START = "10" TYPE = "a"> // start attribute takes a numeric value to begin with the list numbering
TYPE attribute sets 'a' for lowercase letters, 'A' for uppercase, 'i' for lowercase roman, 'I' for uppercase roman numerals. The number given to start attribute indicates which position of number or alphabet it needs to start from.

<UL TYPE = "square"> // unordered list attribute type will use bulleting symbol as circle, square for indication of a bullet. .

HORIZONTAL RULE // used for styling the horizontal rule tag with attributes like <HR ALIGN = "LEFT" WIDTH = "50%"> HEIGHT can also be used. NO SHADE draws horizontal rule without a shadow. SIZE sets vertical size of the rule in pixels. </HR>

TABLE // table element have attribute like FRAME that specifies outside of table with its values like ABOVE, BELOW, HSIDES, VSSIDE, LHS, RHS, VOID and BORDER.

RULES attribute defines rules that appear in the inside of cells it have value GROUPS which display line in grouping of table or NONE, ROWS, COLS, ALL can be used.

LINKING AND ADDRESSING

THE ANCHOR ELEMENT

`<A>` needs two pieces of information: the URL of the target resource, and document content needed to activate that resource. Its `HREF` attribute specifies the target resource ``

` link content text `

Syntax: ` text `
 specifies the url of web `target` specifies where to open page to be linked a linked web page.

``

Bookmark Hyperlink: Go to top or bottom

`<h2 id = "top"> Topic </h2> <p> detail </p>`

` go to top `.

Email hyperlink: To send mail on the mail id

` Send mail `

Contact Number hyperlink: For contact number

` call us `

`<main> click me `

`// target="main" > opens website in new tab.`

`<hav>` used for navigation, unlimited such tags can be used in page `</hav>`

`<hav> home `

` login </hav>`

HTML & MEDIA

HTML Supports : Image Element, Audio element, Video element, I frame element.

Image `` alternate text.

Caption `<figure>` img link inside
`caption → <figcaption> </figcaption>`
`</figure>`

Audio : `<audio src="audio.mp3" controls="controls"></audio>` attribute

Content between audio tags will be shown by browsers which do not support audio. Attributes of audio tags are : loop any value sets to its true type = boolean , executes audio indefinitely.

Autoplay any value set to true [boolean type]

preload : non preloading metadata type specifies whether audio should be preloaded or not.

Muted : any value set to true boolean type mutes

Video : `<video src="myvideo.mp4" control="controls"></video>`

Attribute	Value	Description
loop	Boolean	loops audio indefinitely.
autoplay	Boolean	Plays audio indefinitely.
preload	Metadata	specifies preload or not
height	Pixel	height
width	Pixel	width.
poster	Url of img.	displays img until vdo starts
muted	Boolean.	mutes.

Iframe `<iframe src="url" width="n" height="n"></iframe>`
` click `
`<iframe name="myframe"> </iframe>`

includes document, interactive vdo etc from external src

Block Element

- Begins with a new line
- occupies entire width of parent tag.

Inline Element

- occupies necessary space to accommodate the element.
- elements can be nested with other inline element as well as block elements

Block Elements

Start on a newline
Arrangement

h1

h3

form

div

p

Ex: h1 to h6, p, div,
form.

HTML ELEMENTS

Inline Elements.

Starts on the same
line (if space is available)

Span		a
img.		
Span		a

Ex: a, img, span.

Container Elements.

Start tags <> and End tags </>

Ex: <title> and </title>

Empty Elements.

Stand alone type or self closing tags

Ex:

Unique Elements.

They can be used only once

Ex: <html>, <head> and <body>.

Syntax of <meta> tag : The meta data element is defined within , head tag.

```
<head>
  <meta attributes>
</head>
```

Attributes of Meta.

Attributes	Value	Description
name	application - name description generator	description specifies name description name for the meta.
http-equiv	content type default style refresh	provides http header for information / value of content attribute.
content	text	Gives value associated with http equiv or name attribute.
charset	character set.	Specifies character coding for html document.

```
<head><title> title </title><meta name = "description"
content = " this is a webpage " ./></head>
```

meta provides data about data ; that is not displayed
 specifies page description, author, last modified,
 search engine keywords and web services.

Best Practices

- Title element can also be considered as a part of metadata. Any web page shall only have one title tag per page. Title shall be unique for each page in application.
- Starting title tag with main keyword
- Specify character encoding of document
- Use UTF-8 encoding while designing a webpage.
- Avoid duplicate description inside metadata and try to include the targeted keyword in description inside metadata as search engines index pages based on description.
- Security policy by specifying its value with relevant details. This is to update the browser to load the scripts to be loaded from the trusted origins only.

Therefore help in preventing security attacks such as cross site scripting.

<meta http-equiv = "content-security-policy" content = "default-src 'self' http://xyz.com" // display on different screen accordingly with: <meta name = "viewport" content = "width = device-width, initial-scale = 1.0">

- Use of content security policy (CSP)
- Implementation of CORS policy for allowing access to application services.
- Sanitization of user inputs.
- Make sure to add "sandbox" attribute to an iframe feature of HTML 5.

Form <Form> // used for creating a form </form>
Attributes: Method defaults to http "get" method of submission to the server. To use http "post" use **method = "post"**

Action: The URL to which the form data has to be submitted.

Target: Specifies if the submitted result will open in current window or a new tab, or on new frame

Syntax: form name "name of form" action =
used for accessing form specifies server side
data by scripting language program that will be
executed when form is
submitted.

"link to server side program" **method = "HTTP Request method"** > <!-- All form elements come here-->
</form> specifies HTTP request method that will be used to submit form data to be used to submit form data to server side program.

Input: Used for collecting data from user.

Syntax: <input type "input type" value = "element value"> specifies type of element.

Possible Values of type : text, password, checkbox, radio, button, submit, reset, image, file, hidden, email, number, range, search, url, colour, date, month, week, etc.

<input type = "text" value=""> // inputs text.

<input type = "password"> // inputs password.

<input type = "radio" name = "gender" checked value = "male"> male

<input type = "radio" name = "gender" checked value = "female"> female.

Gender : male female

<input type="hidden" name="language" value="English"> selects language input.

<input type="color"> defines color picker.

<input type="date"> inputs date.

<input type="datetime-local"> defines time.

<input type="week"> defines week.

<input type="month"> defines month.

Button: submit default with button tag.
reset to reset the form
button just click able.

<textarea> defines a multiline field.

<textarea rows="4" cols="10"> default value. </textarea> area space of 4 rows & 10 columns that will input text that enters.

<label> element associated with a text label to tell what kind of input we want

<label for="username"> enter username refers to id attribute of an input element.

</label> <input type="text" id="username">

<Select> defines a dropdown list.

"multiple" attribute can be used for having a multiselect dropdown menu.

Country code : <select> <option value="">

-- please choose country code --> </option>

<option value="+91"> +91 </option>

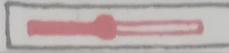
<option value="+41"> +41 </option>

<datalist> predefined elements to choose input

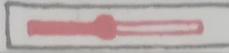
Country <input list="countries"> <datalist id="countries">

<option value="India">

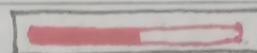
<option value="USA"> </datalist>

Select allows user to select from some predefined options, whereas datalist element even though it is suggested to select from the given option the user can actually enter the data to the input field as  any other input field.

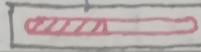
volume: <input type = "range" />

volume 

metre: Disk usage <input min="0" max="100" value="50" /> </input>

Disk usage 

progress: Task complete: <input min="0" max="100" value="50" /> 50 of 100 </input>

task complete 

Output: displays output of user input.

<form oninput = "result.value = parseInt(a.value) + parseInt(b.value)">

<input type = "number" id = "b" name = "b" value = "10" /> +

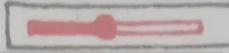
<input type = "number" id = "a" name = "a" value = "13" /> =

Output name = "result".for "a b" > 20 <output>

</form>  +  = 23.

Attributes of input element: placeholder, pattern, min, max, step, required, multiple, form-overide.

Division <div> element is used to group these HTML tags this helps organising web page into diff sections CSS written inside will act upon that block </div>

Span is also used to group element inline. This tag is preferred only when we cannot use  any other semantic tags.

Unordered list // each item within the list is technically referred to as list-item enclosed within `` tag. ``
 Nesting of unordered list.

``

` Markup`

``

` basics of html`

``

` first level `

``

``

` Adaptive html `

``

• **Markup**

• **Basics of HTML**

• **first level**

□ **styling.**

• **CSS**

• **latest version.**

Ordered list

` ordered list `

using CSS `<ol style = "list-style-type: upper-roman;">`

` list `

` list2 `

I. list

II. list2.

attributes:

start: starts the list

reverse: prints the list reversed.

Descriptive list. <dl>

```
<dt> descriptive term </dt>
<dd> descriptive definition </dd>
<dt> descriptive term 2 </dt>
<dd> descriptive definition 2 </dd> </dl>
```

Title

contents

Title 2

contents

Quotation Element.

```
</body>
```

below line is from this website : <blockquote
cite = "https://website.com..

The quotation </blockquote>

```
<Table> <!--table clata --> </table>
```

Elements of table element.

Caption: defines table heading.

tr : table row.

th : heading of column.

td : data of column.

tbody : content part of table.

colgroup: logical group two or more column.

// elements <td> and <th> support attributes namely colspan and rowspan which helps merge the table cells accordingly. The colspan accepts a numerical value and merges specified numeric value of columns together whereas the rowspan attribute accepts a numeric value and merges specific value and merges of rows together.

`<tr> <td colspan = "2"> A </td>`

`<td> B </td>`

`<td> C </td> </td> </tr>`

	C1	C2	C3	C4
R1				
R2				
R3				
R4				

Rowspan: `<tr> <td rowspan = "2"> A </td>`

`<td> B </td>`

`<td> C </td> </td> </tr>`

C1 C2 C3 C4

R1				
R2				
R3				
R4				

Vulnerabilities.

- HTML injection. attacker injects malicious HTML code. can be done through input fields or website links. Happens when user input is not properly sanitized or output is not properly encoded.
- `document.write()` method is used to inject malicious code, used mostly for input fields like comment box, registration forms, etc.
- Stored html injection stores in backend
- reflected html injection stores in web server.
- Prevention Measures.
- use of safe javascript methods like `innerText` in place of `innerHTML`.
- code sanitization : remove illegal character input & output.
- output encoding : untrusted data → safe data where data renders to user without execution converts special character in input and output to unexecutable entities.
- Clickjacking. involves overlaying a form field or something upon the webpage so that user might enter data in wrong malicious field instead of the one they are supposed to.
- done mostly using iframes with low Opacity or transparent layers.
- UI redressing or iframe overlay.
- Prevention.
- Frame buster or frame killer, can be bypassed but it is effective.
- It ensures that current frame is the top level window.

- Using X-frame-response header.
- Using Content Security Policy header.
- X-frame is a response header to be set as a part of http request response. It specifies whether a browser permits to show a webpage inside a <frame> or <iframe> tag.
values : Sameorigin permits frame display on other page but only current domain.
Deny - does not permit any domain to render current page . Allow - from URI allows current page to be displayed in frame but only in specific Content Security Policy. is an HTTP response header providing different directives that limit how & where content is loaded. A csp compatible browser will execute only the scripts in the allowed list of domains. To configure csp we need to set "Content Security Policy" response header or in <meta>

Syntax : Content-Security-Policy : frame-ancestors 'none' doesn't allow any domain to embed a web Content-Security-Policy : frame-ancestors 'self' allows only current page embedded in frame content-Security-Policy : frame-ancestors 'self' '* .example.com' 'https://website.com'; allows current page , any page from website.com to be embedded in frame.

- If iframe needs to be used it shall be used in sandbox to prevent crackjacking.
- Local Storage vulnerabilities.

Data can be stored in two ways : local storage and session storage [they store hold data in key value pairs]

local storage holds data in browser cache until deleted by user or expires on given date. SetItem() is used to assign data to local storage. Syntax:

localStorage.setItem("bgcolor", document.getElementById("bgcolour").Value); creates item with background colour. Get Item() is used to fetch values from local storage.

Syntax: document.getElementById("page").style.backgroundColor = localStorage.getItem("bgcolour");

Session storage holds data until the session ends or the browser tab is closed.

An attacker can inject some malicious code and can steal the data stored here.

So sensitive information is not stored at the client side.

Use cookies with the 'httponly' flag to protect the data stored at the client side.

Cookies it is a piece of data stored in user's browser. It can store upto 4kb of data in browser. Data stored in key value pairs. Syntax: document.cookie = cookieValue;

Set-Cookie:<name>=<value>[;<Max-Age>=<age>]
[; expires=<date>][; domain=<domain_name>]
[; path=<some_path>][; secure][; HttpOnly]

Reverse Tabnabbing. It is an attack where the linked page can control the parent page and can rewrite it without the notice of original user. The changed page looks similar to original leading to a phishing attack where attacker can steal authentication data. Attack can happen in two scenarios.

1. If a website is using a hyperlink with target attribute in that a new tab will open.
 2. If website uses a window.open js function.
[a button using a script with window.open]
- Mitigation: If we don't want set opener and noReferrer option for every hyperlink Set HTTP header
1. Referrer-Policy: no-referrer.
 2. ` click here to win `
 3. `var newWindow = window.open(null, name, 'noopener, noReferrer, ' + options);`