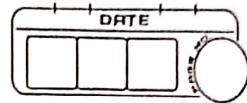


HTML



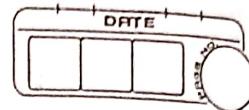
HTML = HyperText Markup Language

#	client side	server side
execution location	- executes on user device	- executes on remote machine
languages	Primarily JS, HTML, CSS	- PHP, Python, Java, etc
main job	- makes clicks & scroll works	- manages saved info
Access level	can't access server data directly	- can read/write files, interact with database
speed	quicker for UI changes	- slower due to network latency

HTML, CSS, JS $\xrightarrow{\text{download}}$ CROME $\xrightarrow{\text{Rendering}}$ WEB PAGE

HTML :-

- 1) Structure :- sets up the layout
- 2) Content :- Add text, images, links
- 3) Tags :- uses elements like `<P>`, `<U>`
- 4) Hierarchy : organizes elements in a tree



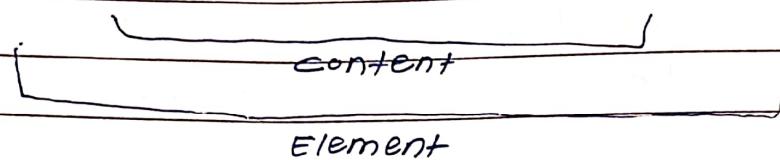
Basics of HTML :-

* What are the tags ?

opening tag

- closing tag

<P>My cat is very grumpy </P>



- elements that are used to create website
- Tags can contain content or other HTML tags.
- use to define elements like text, images, link.

* Using Emmet ! to generate code :-

* Basic HTML page

<!DOCTYPE html> → define the html version

<html lang="en"> → Parent of all HTML tags

<head> → Parent of meta tag

<title> My First Webpage </title> → Title of
webpage

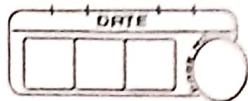
</head>

<body> → Parent of content tag

<h1> Hello world </h1> → heading tag

</body>

</html>



* MDN Documentation :-

- <https://developer.mozilla.org/>
- official resource for HTML

* Comments :-

- Syntax :

' <!-- comment --> '

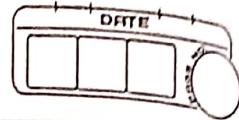
- used to add notes in html code
- helpful for code organization
- can be multiline or single line

* Case sensitivity :

- HTML is not a case-sensitive for html tags
- Attribute names are also case-insensitive
- use lowercase for consistency

Project Level 1 :-

- create a new project with index.html
- generate boilerplate code using emmet
- write "I am learning with Prashant sir"
- use comments
- ALSO use case insensitive tags



HTML Attribute :-

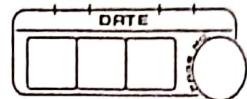
* What are the HTML attribute :-

<tag attribute = "value" > Text content </tag>

- use to provide additional info about element
- Placed within opening tag
- common eg. href, src, alt
- use name = value format
- can be single or multiple per element

* id property :-

- unique identifier :- each id should be unique within a page
- Anchoring :- Allow for direct links to sections using the #id syntax in URLs
- CSS & JS :- used for selecting elements for styling or scripting.



HTML Tags :-

* Heading tag :-

- defines heading in a document
- Ranges from `<h1>` to `<h6>`
- `<h1>` is most important, `<h6>` is least
- important for SEO
- helps in structuring content

* Paragraph tag :-

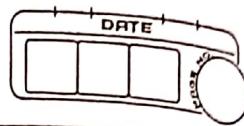
- used for defining tags paragraphs
- enclosed within `<p>` & `</p>` tags
- adds automatic spacing before and after
- Text wrap to next line inside tag
- commonly used in text heavy content

* `
` tag :-

- `
` tags add a line break within text
- `
` is empty, no closing tag needed
- `
` & `
` both are valid

* `<hr>` :-

- It creates a horizontal line
- `<hr>` also empty, acts as divider



* Image Tag :-

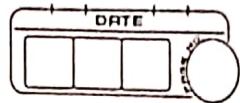
```
<image src="link"  
alt="description/alternative text"  
height="300px">
```

- used to embed images
- utilizes the src attribute for image URL
- alt attribute for alternative text
- can resize using height & width
- self-closing, doesn't require end tag

* Video Tag :-

```
<video src="link"  
height="30px"  
autoplay  
loop  
muted  
controls>
```

- embeds video files on a page
- uses src attribute for video URL
- support multiple format like MP4, WebM
- Allow for built in controls via attributes like autoplay, controls, loop, etc.



* Anchor Tag / hyperlink :-

```
<a href="URL"  
target ="_blank">  
website name to open link  
</a>
```

- link to anything else
- used for creating hyperlink
- use href attribute for URL
- use to link to external site or internal pages
- use target = "_blank" attribute to not open in main / same tab.

* bold :-

- makes text bold

* Italic :-

- <i> makes text italic

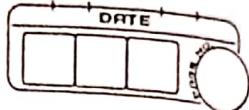
* Underline :-

- <u> ~~not~~ underlines text

* Strikethrough :-

- <s> or <strike> applies strike through

- All above four primarily used for text styling and emphasis



* Pre Tag :-

<Pre>

text

</Pre>

- preserve text formatting
- maintain white spaces & line break
- useful for displaying code
- enclosed within <Pre> and </Pre>
- e.g. display letter, emails, etc.

* Big / Small Tag :-

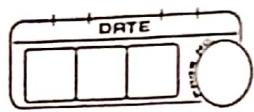
- <big> increases text size
- <small> decreases text size
- less common due to CSS alternatives

* Superscript Tag :-

- <sup> makes text superscript
- used for mathematical equation, footnotes
- does not change font size, just position

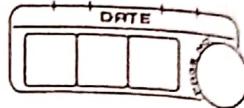
* Character Entity Reference :-

- used to display reserved or special character
- syntax often starts with '' & ends with '''
- e.g. for &



□ Project level 2 :

- 1) create a page with heading , paragraph , line breaks and separators .
- 2) use an image with height 300 , which is a link to another page .
- 3) use bold , italic , underline strike through in one line .
- 4) write third equation of motion using superscript and subscript .



Level 3 - browser Tools

* Browser Tools :-

1] view page source :

- when we click right side of mouse then ~~use~~ there is 'view page source' option OR 'ctrl + U'
- display raw HTML & CSS
- useful for learning & debugging
- show external files like javascript, links

2] Inspect Element :

- Allow real time editing of HTML & CSS
- useful for debugging and testing
- show element hierarchy and layout
- include console of JS, by default
- highlight selected element on page

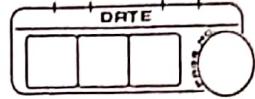
3] HTML without CSS :-

- elements are shown in unorganized manner
- CSS makes page organized, viewable.

* Responsive design :

• different screen sizes :-

- Adapts different layout for different screen sizes.
- Flexible layouts
- optimizes images & assets
- enhances user experience on mobile & desktop



* Live Edit code :-

1] Live edit html :-

- open Kg coding youtube channel & change subscribers no. to 100M

2] Live edit css :-

- open Kg coding youtube channel & change Kg coding color

3] Live edit JS :-

Select element and set visibility to hidden

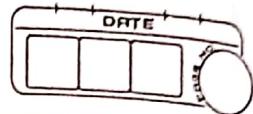
4] changes happening at client :-

- changes made are temp.
- Affect only the current session
- not saved to the server
- reset upon page refresh
- useful for testing, not permanent fixes.

like :- If you change the question in your question paper that has no effect on actual exam.

• Advantages :-

Live edit code provides instant feedback by showing changes in real-time, speeding up development and debugging. It's especially useful for quickly testing and refining UI & front-end code.

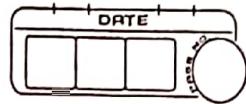


★ Validating webpage :-

- modern browsers are very forgiving
- using validator.w3.org :-
 - Ensures HTML adheres to standards
 - minimizes cross browser issues
 - Helps in achieving better SEO results
 - Easier to debug & maintain
 - optimizes performance & reducing parsing errors

□ Project Level 3 :-

- 1) Save source of Instagram in a file and check the render
- 2) inspect the like element on the page and read the code to understand.
- 3) change no. of likes on your insta post
- 4) validate the project we created in last project



HTML & Project structure :-

* Semantic / non-semantic :-

1) Semantic Tag :-

- Use correct tag for correct tag
- Meaningful : describe content
- SEO : good for search engine
- Accessibility : useful for screen readers
- eg. `<header>`, `<footer>`, `<article>`, `<section>`, `<nav>`

2) Non-Semantic Tag :-

- Generic : no specific meaning
- for styling : used for layout
- NO SEO : Not SEO-friendly
- eg. `<div>`, `<i>`, ``, ``

* Body Tag :-

1) Header Tag :-

- Purpose : used to contain introductory content or navigation links
- semantic : It's a semantic tag, providing meaning to enclosed content
- Location : commonly found at top of page, but can also be used in `<article>` or `<section>` tag
- Multiple Instances : can be used more than once on a page

2) Main tag :-

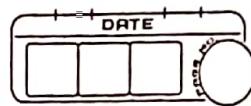
- Purpose : encloses the primary content of a webpage
- Semantic : Adds meaning, indicating the main content area.
- Unique : should appear only once per page
- Accessibility : Helps screen readers identify key content.
- Not for sidebars : Excludes content repeated across multiple pages, navigation or footer.

i) SECTION TAG :-

- Purpose : groups related content in a distinct section.
- Semantic : Adds structure & meaning
- Headers : often used with a heading `<h1>` to `<h6>` to indicate section topic
- Nested : can be nested within other `<section>` or `<article>`.

ii) Article Tag :-

- Purpose : Encloses content that stands alone, like a blog post or news story
- Semantic : Provides contextual meaning
- Independence : content should make sense even if taken out of page context
- Multiple Instances : can be used multiple times on the same page.



iii) Aside Tag :-

- Purpose : Contains sidebar or supplementary content
- Semantic : indicates content tangentially related to main content
- Not crucial : content is not essential to understanding the main content
- e.g. could hold widgets, quotes or ads.

3] Footer Tag :-

- Purpose : For content like extra info or links
- semantic : Provides meaning to enclose content
- Location : Typically at the bottom of page or sections
- content : include copyrights, contact info & social links
- Multiple instances : can be used more than once on a page

* Folder Structure :-

- 1) Root Directory : main folder containing all websites files.
- 2) HTML files : Store main.html files at the root level for easy access.
- 3) CSS folder : create a css folder for all cascading style sheets
- 4) JS files : use a scripts folder for Javascript files.
- 5) Image Folder : store images in an images folder

- c) Assets: other assets like fonts, videos can go in assets/folder
- d) sub-directories: for multipage web-site, use sub-folders to categorize content

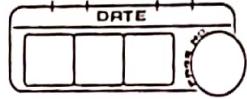
* More Tags :-

1] Navigation Tag :-

- Purpose: Encloses navigation ~~like~~ links or menus
- Semantic: signals that the content is meant for navigating the site
- common content: usually contains lists ``, `` of links `<a>`
- Aids
- Accessibility: Aids screen readers in identifying site navigation

2] Block / Inline elements :-

Block Element	Inline element
• New Line: start on a new line	• Flow: stay in line with text
• Full width: Take up all horizontal space	• width: just as wide as the content
• styling: can have margins & padding	• NO Break: No new line bet ⁿ elements
• size: width & height can be set	• Limited styling: can't set size easily
• example: <code><div></code> , <code><p></code> , <code><h1></code> , <code></code> , <code></code>	• eg: <code></code> , <code></code> , <code></code> , <code><a></code> , <code></code>



* DIV Tag :-

- It is a generic tag
- It is a grouping of tags
- Purpose: Acts as a container for other HTML elements.
- Non-semantic: doesn't provide inherent meaning to enclosed container.
- Styling: commonly used for layout and styling via CSS
- Flexibility: Highly versatile & can be customized using classes or IDs

* SPAN Tag :-

- If we want to create a logical grouping inside the text.
- Purpose: Used for inline elements to define style or manipulate a portion of text.
- Non-semantic: doesn't add specific meaning to the enclosed text.
- Styling: commonly used for changing color, font or adding effects via CSS
- Inline Nature: Doesn't break text flow or create a new block-level element.

D) Project Level 4 :-

- 1) Create a page with Footer, header, main(section, article, aside Tags)
- 2) Make sure Project from level 0 has correct FI structure
- 3) Create a grouping of multiple tags using div
- 4) Create navigation to important sections of your page



4) List Tag :-

1) Ordered Lists:-

- Purpose: used for creating lists with items that have a specific order
- Defaults: Items are automatically numbered.
- Nesting: can be nested within other lists

2) Types of ordered list :-

• Numeric : Default Type (1, 2, 3, ...)

Attribute: type = "1"

• Uppercase Letters : (A, B, C, ...)

Attribute: type = "A"

• Lowercase Letters : (a, b, c, ...)

Attribute: type = "a"

• Uppercase Roman : (I, II, III, ...)

Attribute: type = "I"

• Lowercase Roman : (i, ii, iii, ...)

Attribute: type = "i"

3) Unordered list :-

- Purpose: used for lists where the order of items doesn't matter
- Default: Items are usually bulleted
- Nesting: can be nested within other lists

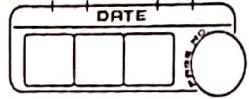


TABLE Tags :-

1) <tr>, <td>, <th> Tags :-

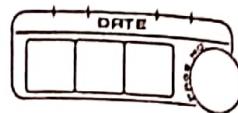
- <tr>: Table Row: used to define a row in an HTML table
- <td> : Table Data: This hold the actual data
- <th>: Table Header: used for header cells within a row. This text is bold & centered by default.

2) caption :-

- Purpose: provides a title or description for a table.
- Placement: Must be inserted immediately after the <table> opening
- Alignment: centered above the table by default
- Accessibility: Helps screen readers to understand the table's purpose

3) column span :-

- Attribute: uses the colspan attribute in <td> & <th> tags.
- Purpose: Allow a cell to span multiple columns horizontally.
- Alignment: Takes the space of the specified number of columns.
- Layouts: useful for combining cells to create complex table layout.



Forms :-

D Input Tag :-

- Purpose : used within a `<form>` element to collect user input.
- self-closing : The `<input>` tag is self-closing doesn't require a closing tag.
- Attributes : common attributes are name, value, placeholder & required.

• Syntax :

```
<form>  
    Text : <input type="Text" Attribute>  
</form>
```

2] Action Attribute :-

• Syntax :

```
<form action="/submit.php" method="Post">  
    <input type="text" name="name">  
    <input type="submit" value="submit">  
</form>
```

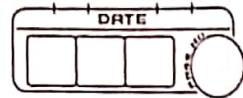
submit કરતોણ ફોર્મ ડેટા ભાગની
URL એ નિર્ધારયા રે specify કરતે

- Purpose : specifies the URL to which the form data should be sent when submitted.

- Default : If not specified, the form will be submitted to the current page's URL.

- Server-side : usually points to a server-side script (like PHP, Python, etc) that processes form data

ID :- Use to target a specific element



3) Name & Value Property :-

- Syntax:

```
<form action="URL" method="Post">  
    <input type="—" name="—" value="—" />  
</form>
```

- Name Property :

- ID for data: It gives a label to the input. It helps server understand what the user entered.

- Unique: each input should have a different/unique name to avoid confusion or form clarity

- Value Property :

- Default data: set initial value for input elements
 - sent to server: This is the data sent when form is submitted.

4) Label Tag :-

- Syntax:

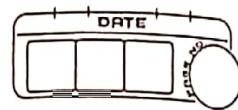
```
<form>  
    <label for="username">Username:</label>  
</form>  
  
<input type="text" id="username" Attribute>  
</form>
```

- Purpose: Adds a text description to form elements.

- For attribute: connects a label to a specific form element using the element's id

- Accessibility: Makes the form more accessible.

- Readability: Enhances form readability & usability



5) Input Tag TYPES :-

a) Input type - DATE :-

- Create a widget to display & pick a date. The browser shows a calendar UI to make selection easy.

- common Attribute:

i) value - set a default date

ii) min - set minimum date

iii) max - set maximum date

iv) required - must choose a date

• Syntax: `<input type="date", id=" " >`

b) input type - FILE :-

- It is used when you want the user to upload a file from their computer or phone.

- It shows a "choose file" or "Browser" button

- useful Attribute:

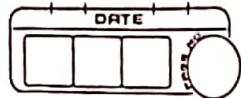
i) accept - limit file type

ii) multiple - allow multiple files

iii) required - must choose a file

iv) name - name of the file input.

• Syntax: `<input type="file" .>`



c) input type - COLOR :

- It creates a color picker. The user can choose a color using a color box or palette
- When you click the box, a color picker appears, and you can select any color.
- Useful attributes:

i) value - set default color

ii) name - used to send data in forms

iii) Required

d) input type - RANGE :

- Syntax :

```
<input type="range">
```

- It creates a horizontal slider that lets the user choose a number between a range

- The user can slide left or right to pick a value.

- Useful Attributes:

i) min - to set minimum value

ii) max - to set maximum value

iii) value - starting value

iv) step - jump between values

v) name - used in forms

e) input type = BUTTON :-

- Syntax :

```
<input type = "button" value = " " >
```

- It creates a clickable button that does nothing by default.

- You usually use it with Javascript to perform an action when clicked.

- It will show a button with the text given in value, but if you want it to do something (like show a message) you need Javascript.

~~- useful attribute~~

f) input type = SUBMIT :-

- Syntax :

```
<input type = "submit" >
```

- It creates a submit button that sends the form data to the server when clicked.

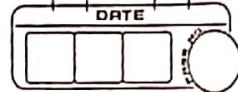
g) Radio :-

- Syntax :

```
<input type = "radio" name = "same-name"  
value = " " >
```

- It is used to let the user choose one option from multiple choices.

- If we want to select only one option then set same name to all elements.



- Attributes :

- i) name - groups radio button together
- ii) value - The value sent if selected
- iii) checked - Makes one option selected by default.

h) input type = CHECKBOX :-

- Syntax :

```
<input type="checkbox">
```

- It lets the user to select one or more option from a list

- Unlike radio buttons , you can select multiple options at the same time.

- useful Attributes :

- i) name
- ii) value
- iii) checked
- iv) id

I) input type = SELECT :-

- Syntax :

```
<select id="" name="">  
  <option value=""> item name </option>  
</select>
```

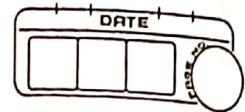
- <select> is used to create a drop-down list

- It lets the user to choose one (or more) options from a list

- <option> tag is used for each item in the list.

- useful Attributes :

- i) name - Name sent with form
- ii) value - value sent when select



- iii) selected - option selected by default
- iv) multiple - Allow selecting more than one option
- v) disabled - makes it unselectable

J) TextArea Tag :-

- purpose : <textarea> is used for multi-line text input in forms.
- rows Property : specifies the visible number of lines in the textarea.
- cols Property : sets the visible width measured in avg character width.
- Resizable : some browsers allow users to manually resize the textarea.

• Syntax :

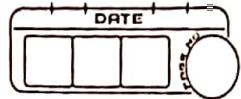
```
<textarea rows="x" cols="y">
```

Default Text..

```
</textarea>
```

• Useful Attributes :

- i) name - name sent with form
- ii) rows - number of visible text lines
- iii) cols - width of the box (characters)
- iv) placeholder - shows hint text
- v) maxlength - limit number of characters
- vi) required - Makes it a must-fill field.



A) iFrame Tag :-

• Syntax:

```
<iFrame width="" height="" src="URL">  
</iFrame>
```

- Embedded content: Allows you to embed another webpage or multimedia content within a webpage.
- Src Attribute: specifies the URL of the content to be embedded.
- Dimensions: width and height can be set using width and height.

B) Project Level 5 :-

- 1) Create a page with all types of ordered list and one unordered list.
- 2) Create a table with headings, captions and a few rows. One of the heading should take at least 3 columns.
- 3) Create a contact me form with relevant details for your resume website.
- 4) Use iFrame to add one movie trailer video & that movie's wikipedia content.