

Step By Step

HTML Notes.

By - KU Coding. + Bonus Level.

Email

- ; BONUS-NOTES

'IMPORTANT HTML CHART SHEET !'

Basic Structure :-

```
<!DOCTYPE html>
<html>
<head>
<title>Title goes here</title>
</head>
<body>
<!--Content goes here-->
</body>
</html>
```

Headings :-

```
<H1> Heading 1 </H1> - Most Important
<H2> Heading 2 </H2>
<H3> Heading 3 </H3>
<H4> Heading 4 </H4>
<H5> Heading 5 </H5>
<H6> Heading 6 </H6> - Least Important
```

Paragraph :-

```
<p>This is my Paragraph</p>
```

Links :-

```
<a href="https://www.HTML.com">A</a>
```

- ; Images ! -

```

```

- ; HTML FORMATTING ! -

 Bold text .
<i> Italic text .
<mark> Highlight text .
 Deleted text . Crossline .
<sup> Subscript | Top .
<sub> Subscript | Bottom .

- ; HTML INPUT TYPES ! -

- ① <input type="text">
- ② Some AS = "password"
- ③ Some AS = "number"
- ④ Some AS = "email"
- ⑤ Some AS = "url"
- ⑥ Some AS = "Date"
- ⑦ Some AS = "time"
- ⑧ Some AS = "datetime-local"
- ⑨ Some AS = "month"
- ⑩ Some AS = "week"
- ⑪ Some AS = "color"
- ⑫ Some AS = "checkbox"
- ⑬ Some AS = "radio"
- ⑭ Some AS = "file"
- ⑮ Some AS = "Submit"
- ⑯ Some AS = "Reset"
- ⑰ Some AS = "Button"
- ⑱ Some AS = "Hidden"
- ⑲ Some AS = "Range"
- ⑳ Some AS = "Search"

-; Special Characters:-

`©` :- Copyright symbol "©"

`&` :- Ampersand character "&"

`<` :- Less-than symbol "<"

`>` :- Greater-than symbol ">"

`"` :- double quotation mark "" "

` ` :- Non-breaking space

Semantic Elements:-

`<header> Header </header>`

`<nau> Navigation </nau>`

`<section> Section </section>`

`<article> Article </article>`

`<aside> Aside </aside>`

`<footer> footer </footer>`

-; Audio:-

`<audio controls>`

`<source src="audio.mp3" type="audio/mp3">`

`</audio>`

-; Video:-

`<video controls width="320" height="240">`

`<source src="movie.mp4" type="video/mp4">`

`</video>`

-; Unordered List:-

``

` ITEM 1 ` - List ITEM

` ITEM 2 `

``

(3)

-: Ordered List :-

```
<ol>  
<li> item1 </li> - List Item  
<li> item2 </li>  
</ol>
```

-: Tables :-

```
<table>  
<tr> - Table Row  
<th> Header 1 </th> - Table  
<th> Header 2 </th>  
</tr>  
  
<tr>  
<td> Data 1 </td> - Table Data  
<td> Data 2 </td>  
</tr>  
</table>
```

-: Comments :-

```
<!-- This is a Comment -->
```

HTML - NOTES

①

Course Level -

- Level 0 Setup8 fundamentals →
- Level 1 HTML Basics →
- Level 2 must-use HTML Tags →
- Level 3 Browser Tools →
- Level 4 HTML and Projects Structure →
- Level 5 List, Tables & Forms →
- Level Bonus GitHub Pages & Code Space →

Level 0 - Setup8 fundamentals.

IDE or Code Editor

- (i) WHAT IS IDE
- (ii) NEED of IDE
- (iii) IDE Selection
- (iv) Installation & Setup
- (v) VSCode Extension.

- 2. Website Components and fundamentals.
 - (i) Client Side vs Server Side.
 - (ii) FrontEnd / BackEnd / FullStack.
 - (iii) Role of Browser.
 - (iv) HTML
 - (v) CSS
 - (vi) JS

→ WHAT IS IDE ?

- (A) IDE Stands for Integrated Development Environment.
- (B) Software Suite that Consolidates Basic Tools Required for Software Developments.
- (C) Center Hub for Coding, finding Problems, and testing.
- (D) Designed to Improve Developer efficiency.

→ Need of IDE ?

- (A) Streamlined Developments.
- (B) Increase Productivity.
- (C) Simplifies Complex tasks.
- (D) Offer a Unified workspace.
- (E) IDE Features,

- 1. Code auto Complete.
- 2. Syntax Highlighting
- 3. Version Control.
- 4. Error checking.

Brainerd Activity .kt

@Composable
fun messageCard(msg: Message) {
Row(modifier = modifier.padding(all = 8.dp)) {
Image(
 painter = PainterResource(R.drawable.android_studio_logo),
 contentDescription = "Profile Picture",
 modifier = modifier.size(45.dp)
)
Spacer(modifier = modifier.width(8.dp))
Column(modifier = modifier
 .background(Color.LightGray).width(150.dp)) {
Text(text = msg.author, color = Color.Black)
Text(text = msg.body, color = Color.Black)
}

Spacer(modifier = modifier.height(10.dp))
Text(text = msg.body, color = Color.Black)

NEXT -

1.3 IDE Selection

1. Sublime Text
2. Atom
3. VS Code
4. GitHub Code Spaces

2.3 Role of Browser

1. Display web page - Turns HTML code into what you see on screen.
2. User Click - Helps you interact with the web pages.
3. Updates Content - Allows changes to the page using JavaScript.
4. Load files - Gets HTML, Images, etc, from the Server.



2.4 HTML (HyperText Markup Language)

1. Structure : Sets up the layout.
2. Content : Add text, Images, Links,
3. Tags : Uses elements like - <p>, <a>.
4. Hierarchy : organizes elements, Tree.

2.5 CSS (Cascading Style Sheets)

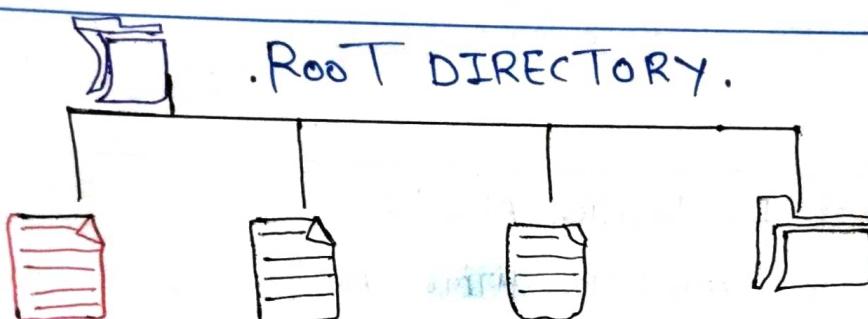
1. Style : Sets the look & feel.
2. Colors & fonts : customize text & background.
3. Layout : controls, Position and size.
4. Selectors : Target Specific HTML Elements.

2.6 JS (JAVASCRIPT)

(2)

1. JAVA Script Has Nothing to do with Java.
2. Action : Enables Interactivity.
3. Updates / Alters Pages without Reloading.
4. Events / Respondes to User Actions.
5. Data / Fetches and Sends info to Server.

- Website Components and fundamentals.
- 1. Client Side vs Server Side
- 2. frontend / Backend / fullstack
- 3. Role of Browser
- 4. HTML
- 5. CSS
- 6. JS



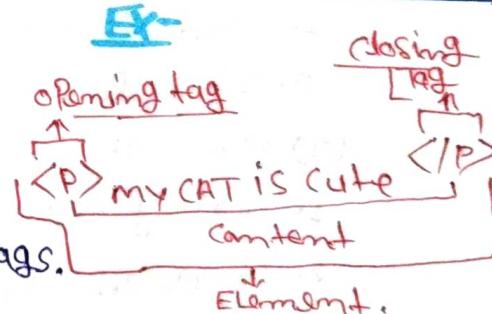
Basic of HTML -

2.1 What are tags?

1. Elements that are used to Create a website are Called HTML Tags.

2. Tags Can Contain other HTML Tags.

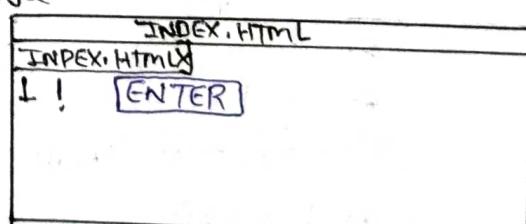
3. Define elements Like text, Img, Links.



2.2 Using Emmet! to Translate Code.

1. TYPE ! and wait for Suggestions.

[2.3 Basic HTML PAGE]



<!DOCTYPE html> → Define the HTML version.

<html lang="en"> → Parent of all HTML Tag / Root element.

<head> → Parent of meta data tags.

<title> My First web Page </title> → Title of the web Page.

</head>

<body> → Parent of Content tags.

<h1> Hello world </h1> → Heading tag.

</body>

</html>

2.5 Comments.

1. Used to add Notes in HTML Code.
2. Not Displayed on the web Page.
3. Syntax: `<!-- Comment Here -->`
4. Helpfull for Code organization.
5. Can be multi-line or single-line.

2.6 Case Sensitivity.

1. HTML is Case-Insensitive for tag Name.
2. Attribute names are also be Case-Insensitive.
3. Best Practice: Use LowerCase for Consistency.

`<html> = <HTML>`
`<p> = <P>`
`<head> = <HEAD>`
`<body> = <BODY>`

- ; Level - 1 Revision ALL ; -

→ Level - 2 -(must use **HTML TAGS**)

(1) **HTML Attributes.**

1. What are HTML Attributes.
2. ID Property.

(2) **HTML TAGS.**

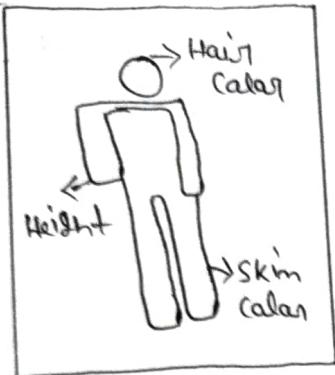
1. Heading Tag
2. Paragraph Tag
3. `
`, `<HR>` tags
4. Image Tag
5. Video Tag
6. Anchor tag
7. Bold/Italic/underline/strikethrough
8. Pre tag
9. Big, Small tag
10. Superscript/Subscript

(3) **Character Entity Reference.**

1. What are Character Entity Reference.

(3)

1. HTML ATTRIBUTES.



1.1 What are HTML Attributes?

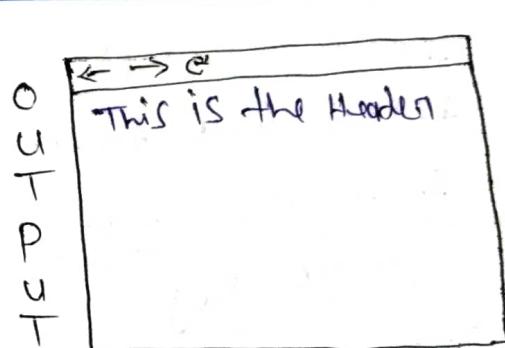
`<tag attribute="value">text Content</tag>`

1. Provides additional info about elements.
2. Placed within opening tags.
3. Common Ex - href, src, alt.
4. Use Name = Value format.
5. Can be single or multiple for element

1.2

ID Property.

- Unique Identifier → Each ID should be unique within a page.
- Anchoring: Allows for direct links to sections using the `#ID` syntax in URLs.
- CSS & JavaScript: used for selecting elements, for styling or scripting.



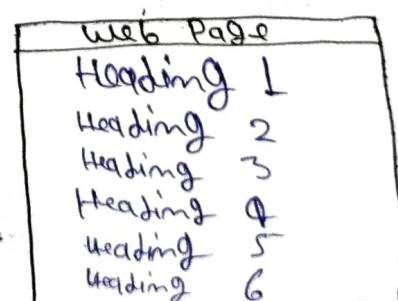
I
N
P
U
T

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with</title>
</head>
<body>
    <div id="Header">This is the Header</div>
</body>
</html>
```

(2. HTML TAGS)

2.1 Heading tag.

1. Define Heading in a Doc.
2. Ranges from `<H1>` to `<H6>`
3. `<H1>` is most important.
- `<H6>` is least.
4. Important for SEO.
5. Helps in structuring content.



CODE →

- HTML Boilerplate

```
<BODY>
<H1> 1 </H1> ] <H6> 6 </H6>
<H2> 2 </H2> ] </BODY>
<H3> 3 </H3>
<H4> 4 </H4>
<H5> 5 </H5>
```

2.2 Paragraph tag.

Same AS,

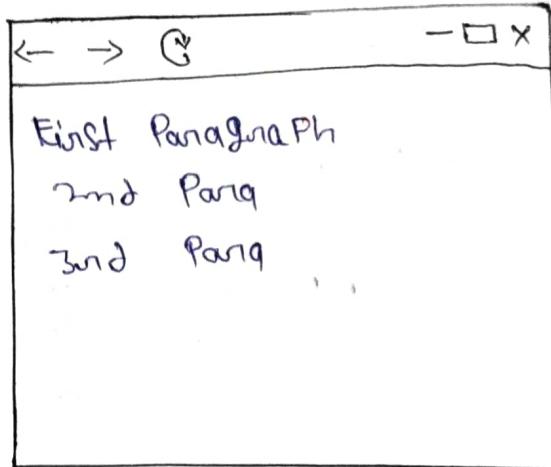
`<Body>`

`<P> first Paragraph </P>`

`<P> 2nd Para </P>`

`<P> 3rd Para </P>`

`</Body>`



1. Used for defining Paragraphs.
2. Enclosed within `<P>` and `</P>` tags.
3. Adds automatic Spacing before and After.
4. Text wraps to Next line inside tag.
5. Common in text - heavy content.

2.3 BR Tag

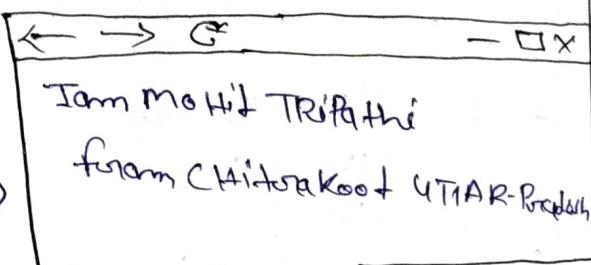
Same AS,

`<Body>`

`<P> Sam mohit Trivarthi </P>`

`
`

`<P> from Chitrakoot UP </P>`



1. `
` adds a line break within text.
2. `
` is empty, no closing tag needed.
3. `
` and `
` are both valid.

2.4 HR Tag

Same AS,

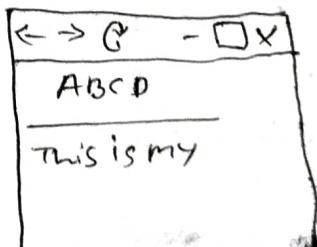
`<Body>`

`<H2> ABCD </H2>`

`<HR>`

`<P> This is my </P>`

`</Body>`



1. `<HR>` creates a Horizontal Rule or line.
2. `<HR>` also empty, acts as a divider.

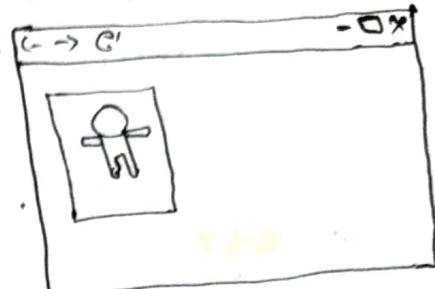
2.4 Image tag.

Same As.

<Body>

<Image src="1.jpg" alt="photo" height="300px">

</Body>



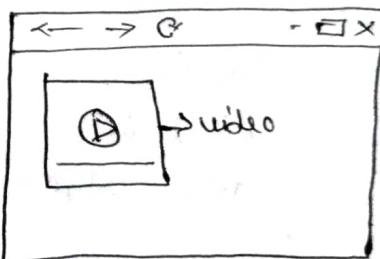
1. Used to embed **Images**.
2. Utilizes the **src** attribute for img url.
3. **alt** attribute for alternative text.
4. Can be resized using **width** and **height**.
5. **Self-Closing**, doesn't require an end tag.

2.5 Video Tag.

Same As, → <Body>

<Video src="1.mp4" height="330px"
autoPlay controls loop/>

<Body>



1. Embeds video file on a page.
2. uses **src** attribute for video url.
3. Supports **multiple formats** like mp4, webm.
4. Allows for built-in controls via attribute like **autoplay**, **controls**, **loop**.

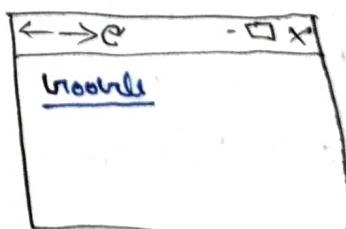
2.6 Anchor tag.

Same As,

<Body>

<a href="https://www.google.com"
target="blank"> Google

<Body>



1. Used for creating **Hyperlink**.
2. Requires href attribute for url.
3. Can link to external sites or internal pages.
4. Support target attribute to control link behavior.

2.7 BOLD/Italic/underline/strikethrough - tag

Same AS.

<Body>

 Bold

<i> Italic </i>

<u> Underline </u>

<s> Strikethrough </s>

<Body>

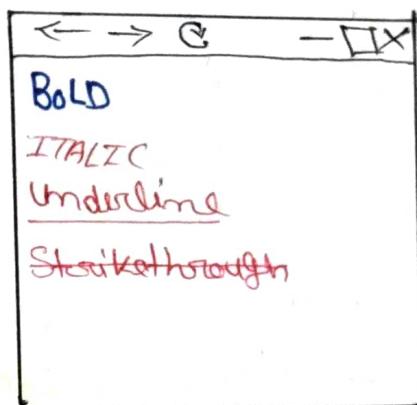
1. makes text **Bold**.

2. <i> makes text **italic**.

3. <u> **Underline** text.

4. <s> or <strike> applies **Strikethrough**.

Primary used for text styling and emphasis.



2.8 Pre tag.

<Body>

first line

more spaces

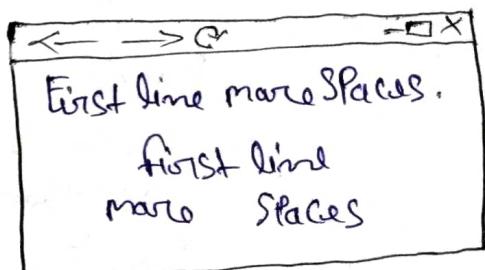
<Pre>

first line

more spaces

<(Pre)>

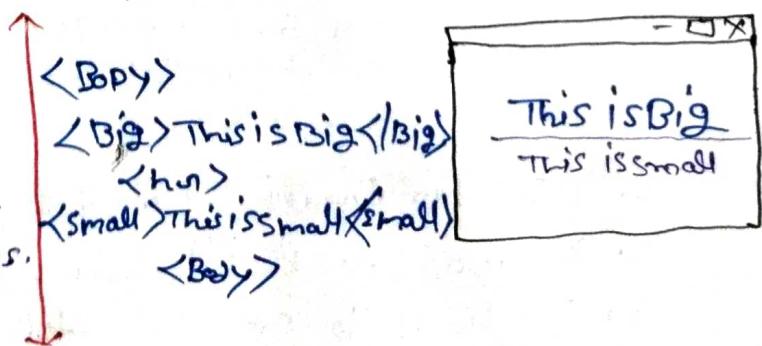
</body>



1. Preserves text **formatting**
2. maintain whitespaces and line breaks,
3. useful for displaying code.
4. Enclosed within <pre> and </pre> tags.

2.9 Big/small Tag

1. <big> increase text size.
2. <small> decrease text size.
3. Less command to CSS alternatives.



(5)

2.10 SuperScript/SubScript Tag.

1. $\langle\text{Sup}\rangle$ makes text Superscript.
2. $\langle\text{Sub}\rangle$ makes text Subscript.
3. Used for mathematical equations
footnotes
4. Does not change font size,
Just position.

```
index.html
<Body>
<Big>  $(a+b)^2 = a^2 + b^2 + 2ab$ 
      </Big> <Hn>
<Big>  $(a+b)^2 = a^2 + b^2 + 2ab$ 
      <Sup>2</Sup> <Sub>+ 2ab</Sub>
      <Big> + 2ab </Big> <Hn>
</Body>
```

$(a+b)^2 = a^2 + b^2 + 2ab$
$(a+b)^2 = a^2 + b^2 + 2ab$
$\text{CH}_4 + \text{O}_2 \Rightarrow \text{H}_2\text{O} + \text{CO}_2$
$\text{CH}_4 + \text{O}_2 \Rightarrow \text{H}_2\text{O} + \text{CO}_2$

Level-2 must-use HTML Tags (3) Character Entity Reference [Attach 3.1 Character Entity Reference]

Level-2 Revision

1. HTML ATTRIBUTES.

1. What are HTML Attributes?
2. ID Property?

2. HTML TAGS?

- | | | |
|--|-------------------------------|---------------|
| 1. Heading Tag | 2. Paragraph Tag | 3. BR/HR Tag |
| 4. Image Tag | 5. Video Tag | 6. Anchor Tag |
| 7. Bold/italic/underline Strikethrough | | 8. Pre Tag |
| 9. Big/Small Tag | 10. Superscript/Subscript Tag | |

3. Character Entity Reference.

1. What are Character Entity Reference?

Level-3 Browser Tools.

1.1 View Page Source.

1. Displays Raw HTML and CSS.
2. Useful for debugging and learning.
3. Shows external files like JavaScript links.

1.2 Inspect Elements.

- 1. Allows real-time editing of HTML/CSS.
- 2. Useful for debugging and testing.
- 3. Shows element hierarchy and layout.
- 4. Includes console for JavaScript.
- 5. Highlights selected elements on page.

IDEAS
CODE

2. Responsive Design.

2.1 Different screen sizes.

- 1. Adapts layout for different screen sizes.
- 2. Flexible layout.
- 3. Optimizes images and assets.
- 4. Enhances user experience on mobile and desktop.

3. Live Edit Code -

3.1 Live edit in HTML. → PDF Page. 65 - 68 line.

4. Validating Web Pages.

Using [Validator.w3.org](http://validator.w3.org)

- 1. Ensures HTML adheres to standard.
- 2. Minimizes cross-browser issues.
- 3. Helps in achieving better SEO result.
- 4. Easier to debug and maintain.
- 5. Optimizes performance by reducing parsing errors.

(6)

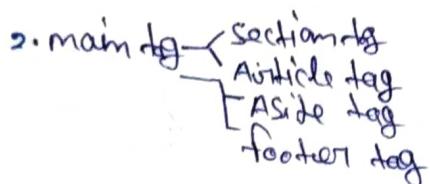
level - 4 HTML Project Structure.

1. Semantic tags.

- semantic / Name semantic tag.

2. Body tags.

1. Header tag



3. folder structure.

1. Recommended Folder Structure

4. more tag.

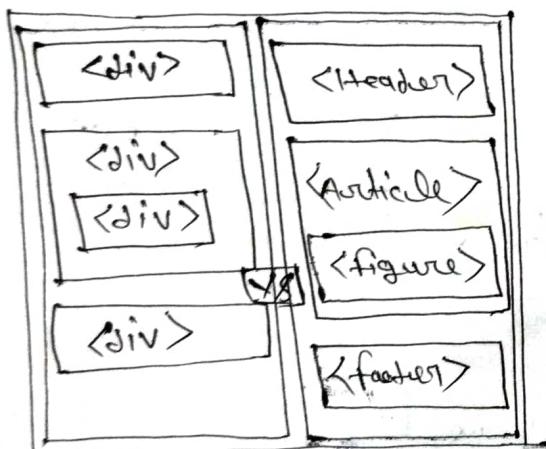
1. Navigation tag

3. Div tag

2. Block/Inline Elements.

4. Span tag

① Semantic tag



①.1 Semantic And Non Semantic tags.

Semantic tags.

- meaningfull**: describe content.
- SEO**: good for search engine.
- ACCESSIBILITY**: useful for screen readers.
- Example**: <header>, <footer>, <article>, <section>, <nav>.

NON Semantic tags.

Generic: No specific meaning.

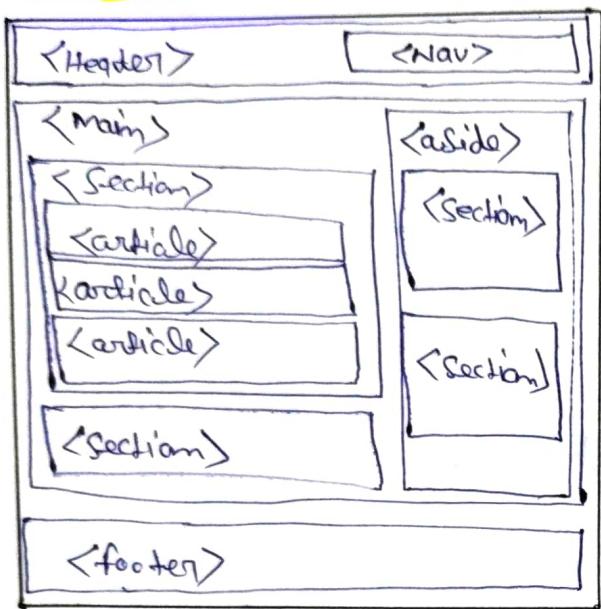
for styling: used for layout.

no SEO: Not SEO friendly.

Example: <div>, , <i>,

Level - 4.

2. Body tags



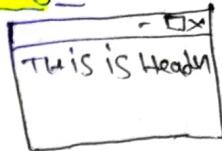
2.1 - Header tag

Same-as.

<Body>

<Header> (big)

This is the Header </big> </Header>
<body>



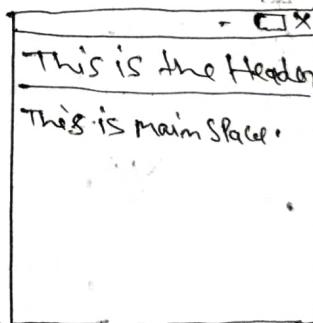
1. **Purpose:** Used to contain introductory Content or Navigation links.
2. **Semantic:** It's a Semantic tag, Providing meaning to the enclosed Content.
3. **Location:** Commonly found at the Top of web pages but can also appear within <article> or <section> tags.
4. **multiple instances:** Can be used more than once on a Page within different Sections.

2.2 Main tag

Same-as.

<Body>

<Header> (big) This is the Header </big> </Header>
<H1> <main> This is the main place </main>
</Body>



1. **Purpose:** Enclose the Primary Content of a web page.
2. **Semantic:** Adds meaning, indicating the main Content area.
3. **unique:** Should appear only once per page.
4. **Accessibility:** Helps Screen Readers identify key Content.
5. **Not for Side Bars:** Excludes Content Repeated across multiple Pages like Side Navigation or footer.

2.2.1 Section tag.

Same As.

`<Body>`

`<main>`

`<Section> Technologies: HTML/CSS </section>`

`<Section> Courses: OS/DBMS </section>`

`<Section> Hobbies: Eating </section>`

`</main>`

`</Body>`

web.com. - □x

Technologies: HTML/CSS

Courses: OS/DBMS

Hobbies: Eating

1. **Purpose**: Groups Related Content in a distinct Section.
2. **Semantic**: Adds Structure and meaning.
3. **Headers**: often used with a Heading `<H1>` to `<H6>` to Indicate Section Topic.
4. **Nested**: Can be Nested within other `<Section>` or `<Article>` tags.

2.2.2 Article tag =>

CODE =>

Same - AS,

`<Body>`

`<main>`

`<article>`

`text ---`

`---`

`</article>`

`</main>`

`</body>`

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 the Page Context.

multiple Instances: Can be multiple times on the same Page.

web Page - □x

Text

2.2.3 Aside tag =>

Purpose:

Contains Sidebar or Supplementary Content.

Semantic:

Indicates Content tangentially Related to the main Content.

Not Crucial:

Content is Not essential Under-standing the main Content.

Example:

Could Hold **widgets, quotes, an ads.**

Same AS,

`<Body>`

`<main>`

`<aside>`

`<h3> Recommended Books`

`</h3>`

``

`....`

`....`

``

`</aside>`

`<main>`

`</body>`

web - □x

Recommended Books

• " " "

• " " "

2.3 Footer tag.

Purpose:

For footer content like footer info or links.

Semantic:

Provides meaning to enclosed content.

Location:

Typically at the bottom of pages or sections.

Content:

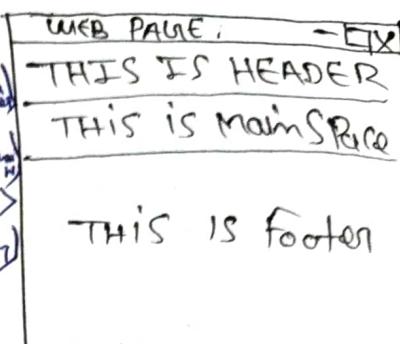
Including copyright, contact info, and social links.

multiple Instances:

Can be used more than once on a page.

Same AS.

```
<Body>
  <Header> This is Header </Header>
  <H1>
    <Main> This is main content </Main>
  <Footer> This is footer </Footer>
</Body>
```



Level-4 3. Folder Structure.

3.1 Recommended folder structure.

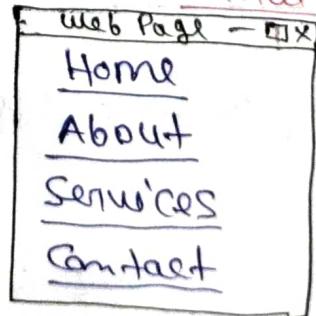
1. Root Directory: main folder containing all website files.
2. HTML Files: Store main HTML files at the root level for easy access.
3. CSS folder: Create a CSS folder for all cascading style sheets.
4. JS Folder: Use a Script folder for JavaScript files.
5. Image folder: Store images in an Images or Images folder.
6. Assets: Other assets like fonts can go in an Assets folder.
7. Sub-directories: For multi-page websites use sub folder to categorize content.

Level-4. 4 more tags.

4.1 Navigation tag.

- Purpose: Encloses navigation links as menu.
- Semantic: Signals that the content it means, ~~for~~ for navigating the site.
- Commonly content lists (`ul`, `ol`) of ~~the~~ links (`a`).
- Accessibility: Aids screen readers in identifying site navigation.

Output



Code area

Same AS.

```
<Body>
  <Nav>
    <a href="#"> Home </a> <br>
    <a href="#"> About </a> <br>
    <a href="#"> Services </a> <br>
    <a href="#"> Contact </a> <br>
  </Nav> </Body>
```

4.2. Block / Inline Elements.

Block ELEMENTS

- New line: Start up a new line.
- full width: Take up all Horizontal space.
- Styling: Can have margin and padding.
- Size: width and height can be set.
- Ex: <div>, <p>, <h1>, ,

INline ELEMENTS

flow: Stay in line with text.
width: Just as wide as the content.
NowBreak: No newline between elements.
limited styling: Can not set size easily.
Ex: , <a>, , , ,

1. Purpose - Acts as a container for other HTML elements.
2. Non-Semantic - Doesn't provide inherent meaning to enclosed content.
3. Styling - Commonly used for layout and styling via CSS.
4. flexibility - Highly versatile and can be customized using classes or IDs.

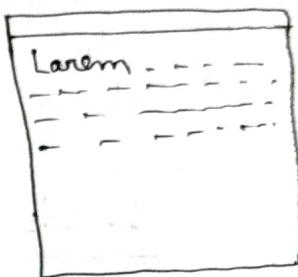
4.3 DIV Tag

Same AS -

<Body>
<div>

<P> Larrem aa </P>

</div>
</Body>



4.4 SPAN Tag

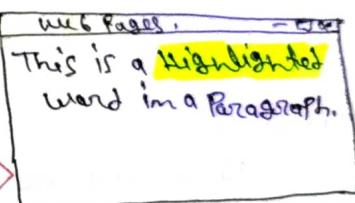
- Purpose: Used for inline elements to style or manipulate a portion of text.
Non-Semantic: Does not add specific meaning to the enclosed text.
Styling: Commonly used for changing color, font, or adding effects via CSS.
inline nature: Does not break text flow or create a new block level element.

Same AS,

<Body>
<P> This is a

class="highlight">

Highlighted word in a Paragraph. </P>
</Body>



Level - 4 Revision

1. Semantic tags, (i) Semantic/Non Semantic tags.
2. Body tags, (i) Header tag (ii) main tag (iii) section tag, Article tag, Aside tag (iv) footer tag
3. folder Structure (i) Recommended folder Structure.
4. More tags,
 - (i) Navigation tag
 - (ii) block/inline elements,
 - (iii) Div tag
 - (iv) Span tag.

Level - 5

List, Table, form.

① List tag -

- (i) ordered list.
- (ii) types of ordered list.
- (iii) unordered list.

② TABLE Tag .

- (i) <tr>, <td>, <th> tags
- (ii) colspan
- (iii) Col Span

③ Forms -

- (i) Input tag
- (ii) Action Attributes
- (iii) Name And Value Property.
- (iv) Label tag
- (v) Selecting Types.

④ I frame Tag .

- (i) Using I frame.

1. List tag .



1.1 ordered List .

1. Refuse -

used for creating list with items that have a specific order.

2. Default : Item are automatically numbered.

3. Nesting : Can be nested within order list.

Some As,

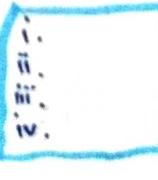
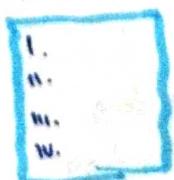
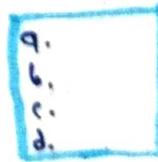
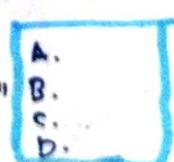
```

<Body>
<ol>
<li> wake up</li>
<li> Brush teeth</li>
<li> Take a Shower</li>
<li> Have Breakfast</li>
<li> Go to work</li>
</ol>
</Body>
```

TYPES OF ordered List

ordered li -

- **Numeric** : Default TYPE, (1,2,3...) Attribute TYPE = "1"
- **Upper Case Letters** : (A,B,C,...) Attributes : TYPE = "A"
- **Upper Case Roman** : (I,II,III,...) Attributes : TYPE = "I"
- **Lower Case Roman** : (i,ii,iii,...) attributes TYPE = "i"



1.3 Unordered Lists.

1. Purpose: - Used for lists where the order of items doesn't matter.
2. Default: - Items are usually bulleted.
3. Nesting: - Can be nested within other lists.

```
<body>
  <ul>
    <li>APPLE</li>
    <li>Banana</li>
    <li>Cherry</li>
    <li>Dragonfruits</li>
  </ul>
</body>
```

•	APPLE
•	Banana
•	Cherry
•	Dragonfruits

Level .5 - 2. TABLE TAG

2.1 <tr>, <th>, <td> TAGS.

1. **<tr>** Table Row: Used to define a row in an HTML table.
2. **<th>** Table Header: Used for header cells within a row, text is bold and created by default.
3. **<td>** Table Data: This holds the actual data.

Same AS,
<body>
<table border="1">
 <tr>
 <th> Name </th>
 <th> Age </th>
 <th> Email </th>
 </tr>
 <tr>
 <td> John </td>
 <td> 30 </td>
 <td> --- </td>
 </tr>
 <tr>
 <td> Jane </td>
 <td> 25 </td>
 <td> --- </td>
 </tr>
</table>
- Same AS, Fill Details,

NAME	Age	Email
John	30	---
Jane	25	---

2.2. CAPTION

To add caption. After **<tr>** tag, before **<table>** tag insert a new tag name: **<caption>** ----- **</caption>**

2.3 Col SPAN.

1. **Attributes:** - uses the **colspan** attribute in **<td>** or **<th>** tags.
2. **Purpose:** - Allows a cell to span multiple columns horizontally.
3. **Alignment:** - Takes the space of the specified number of columns.
4. **Layout:** - Useful for combining cells to create complex table layout.

Same AS,
<body>
<table>
 <tr>
 <td> John </td>
 <td> 30 </td>
 <td> email.com </td>
 </tr>
 <tr>
 <td colspan="3" style="text-align: center;">This cell spans 3 columns

Student Info.		
Name	Age	Email
JOHN	30	email.com
This cell spans 3 columns		

Level - 5 3. Forms.

3.1 INPUT Tag.

1. **Purpose** :- Used within a `<form>` element to collect user input.
2. **Self-Closing** ; The `<input>` tag is self-closing; doesn't require a closing tag.
3. **Attributes** : Common attributes are `name`, `value`, `placeholder` and `required`.

Same As.
<body>
<form>
text : <input type="text"
Placeholder="Enter text">

password : <input type="password">

</form>
</body>

A hand-drawn diagram of a rectangular form. Inside, there is a text input field containing the text "Hello". Below it is a password input field containing the placeholder text ".....".

3.2 Action attribute.

Same As,
<body>
<form action="/submit.php" method="Post">
<input type="text" name="name">
<input type="submit" value="Submit">
</form>
</body>

A hand-drawn diagram of a rectangular form. It contains a text input field and a submit button labeled "Submit".

1. **Purpose** :- Specifies the URL to which the form data should be sent when submitted.
2. **Default** :- If not specified, the form will be submitted to the current page's URL.
3. **Server-side** :- usually points to a server-side script (like PHP, Python, etc.) that processes the form data.

3.3 Name and Value Property.

• Name Property:

- **ID for Data** : Identifier for form elements when submitting.

- **Unique** : Should be unique to each elements for clarity.

• Value Property:

- **Default Data** : Sets initial value for input elements.

- **Sent to Server** : This is the data sent when form is submitted.

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

A hand-drawn diagram of a rectangular form. It contains a text input field with the value "JOHN".

3.4 Label Tag.

1. Purpose: Adds a text description to form elements.
2. for Attribute: Connects the label to a specific form elements using the element's ID.
3. Accessibility: Makes the form more Accessible.
- Readability: Enhance form Readability.

Same As, <body>

```
<form Action="/submit.php">  
Method = "Post">  
<label for="username">Username<br/>
```

A rectangular form with a label 'Username.' followed by a text input field.

```
<input type="text" id="username" Name="username">  
</form>  
</body>
```

3.5 Input Type: Date

for - extra -

File : <input type="file">
 [Output]
 File : No chosen file

Input , Type color -

before text -

Color : <input type="color">
 output
 Color :

Input Type Range :- Range : <input type="range">
 Range :

Input Type : Button .

Button : <input type="button" value="click">
 output -

Button :

Input Type : Submit

Submit : <input type="submit" value="Submit">
 output ;
 Submit

Input Type : Radio .

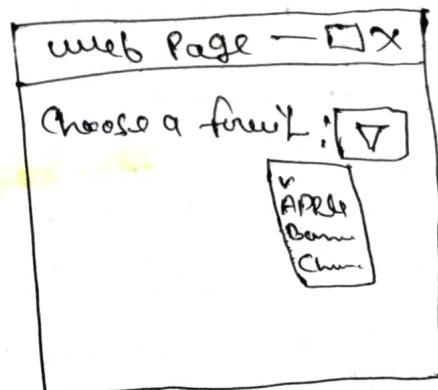
<form>
<label for="male" male> male </label>

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

[male

Input type - Select.

```
<form>  
<label for="favorites"> Choose a fruit: </label>  
<select id="favorites" Name="favorites">  
  <option value="apple"> Apple </option>  
  <option value="banana"> Banana </option>  
  <option value="cherry"> Cherry </option>  
</select>  
</form>  
</body>
```



① Purpose:- <textarea>

is used for multi-line text input informs.

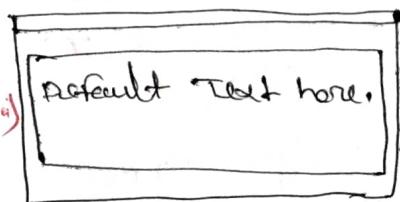
- (i) Rows Property: specifies the visible no. of lines in the textarea.
- (ii) cols Property: sets the visible width measured in Average characters width.

② Resizable:-

Some Browsers allow users to manually Resize the Text area.

Same As,
<Body>

```
<textarea rows="4" cols="40">  
  Default Text Area  
</textare>  
</body>
```



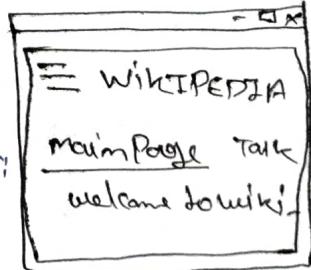
Level - 5 I-frame tag.

4.1 Using I-frame.

Same as,

```
<Body>
```

```
<iFrame width="300" height="200" src="https://en.wikipedia.org/wiki/Main_Page"></iFrame>  
<Body>
```



- ① Embedded Content: Allows You to Embed another Web Pages as multimedia Content within a Web Page.

- ② src Attributes: Specific the url of the content to be embedded.
- ③ Dimensions: width and height can be set using width and height attributes.

- ! Level - 5 Revision :-

① List Tag -

- (i) ordered list
- (ii) types of ordered list
- (iii) unordered list

② TABLE Tag -

- (i) <tbl>, <td>, <th> tags.
- (ii) caption,
- (iii) col \$ Row.

③ Forms -

- (i) Input tag
- (ii) Action Attributes
- (iii) Name & value Property
- (iv) Label tag
- (v) Gathering type

④ IFRAME tag -

- (i) Using Iframe.

Level - Bonus

① GitHub :-

- ① What is version control
- ② What is git & GitHub
- ③ Account creation
- ④ Creating a Repo
- ⑤ Creating a CodeSpace
- ⑥ Creating a GitHub Page
- ⑦ Publishing our Projects.

② Frameworks ,

- ① React
- ② Angular
- ③ Vue

Git - GitHub :

- ① Definition : A System to track changes in files over time.
- ② Types : Centralized (like SVN) and Distributed (like Git).
- ③ Purpose : Helps in teamwork and fixes mistakes.
- ④ Snapshots : each "Commit" saves a file version.
- ⑤ Branching : Lets you work on different tasks separately.
- ⑥ Merge : Combines changes from different people.
- ⑦ Undo : easy to revert to older file version.

1.2 what is git & git hub?

What is git?

- ① Definition: A software tool that tracks changes in code, enabling collaboration and version control.
- ② Commit: Record a snapshot of file changes.
- ③ Branch: Allows separate path of development.
- ④ Merge: Combines changes from different branches.

What is git hub?

- ⑤ Definition: A web service for hosting and collaborating on git repositories.
- ⑥ fork: Creates a personal copy of another user's repository.
- ⑦ Pull Request: A way to propose changes to existing code.
- ⑧ issues: used for tracking bugs and feature ideas.

2. Frameworks