

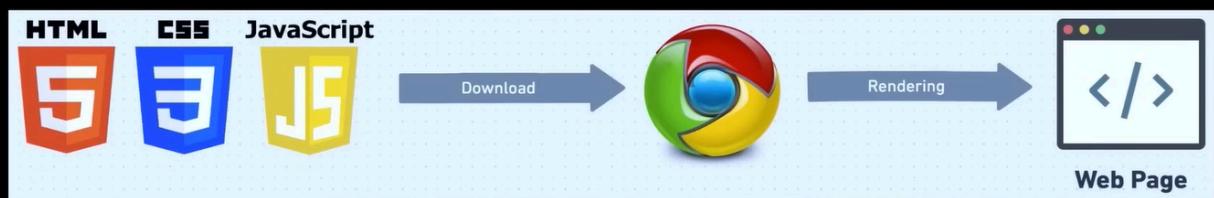
HTML is used for structure and layout of web page and organize elements

## LEVEL 0 2.1 Client Side vs Server Side

	Client Side	Server Side
Execution Location	Executes on user's device.	Executes on a remote machine.
Languages	Primarily JavaScript, HTML, CSS.	PHP, Python, Java, Node.js, etc.
Main Job	Makes clicks and scrolls work	Manages saved information
Access Level	Can't access server data directly	Can read/write files, interact with databases.
Speed	Quicker for UI changes	Slower due to network latency.



## LEVEL 0 2.3 Role of Browser



1. Displays Web Page: Turns HTML code into what you see on screen.
2. User Clicks: Helps you interact with the web page.
3. Updates Content: Allows changes to the page using JavaScript.
4. Loads Files: Gets HTML, images, etc., from the server.



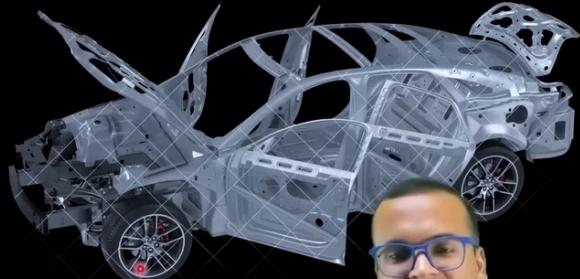
LEVEL 0



## 2.4 HTML

(Hypertext Markup Language)

1. **Structure:** Sets up the layout.
2. **Content:** Adds text, images, links.
3. **Tags:** Uses elements like `<p>`, `<a>`.
4. **Hierarchy:** Organizes elements in a tree.



LEVEL 0



## 2.5 CSS

(Cascading Style Sheets)

1. **Style:** Sets the look and feel.
2. **Colors & Fonts:** Customizes text and background.
3. **Layout:** Controls position and size.
4. **Selectors:** Targets specific **HTML** elements.



**LEVEL 0**



## 2.6 JS

(Java Script)

1. JavaScript has nothing to do with Java
2. **Actions:** Enables interactivity.
3. **Updates:** Alters page without reloading.
4. **Events:** **Responds** to user actions.
5. **Data:** Fetches and sends info to server.



**LEVEL 1**

## 1.2 File Extension

### HTML

1. Most commonly used.
2. Works across all browsers.
3. Widely recognized and supported.
4. Typically saved as .html.



### HTM

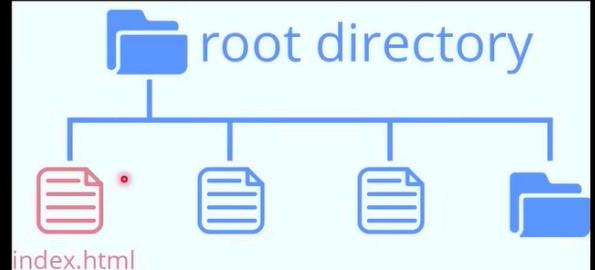
1. Less commonly used.
2. Originated for compatibility with older systems.
3. Works same as .html.
4. Typically saved as .htm.



**LEVEL 1**

## 1.4 Importance of **index.html**

1. Default name of a website's homepage.
2. First page users see when visiting a website
3. Important for **SEO** (Search Engine Optimization)
4. Provides uniform starting point across servers
5. Serves as **fallback** when no file is specified in URL

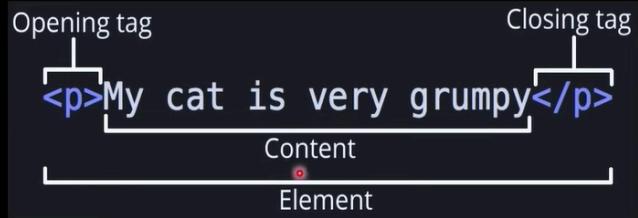


Always store index.html file in root folder, it is opening page so it should load fast

**LEVEL 1**

## 2.1 What are Tags

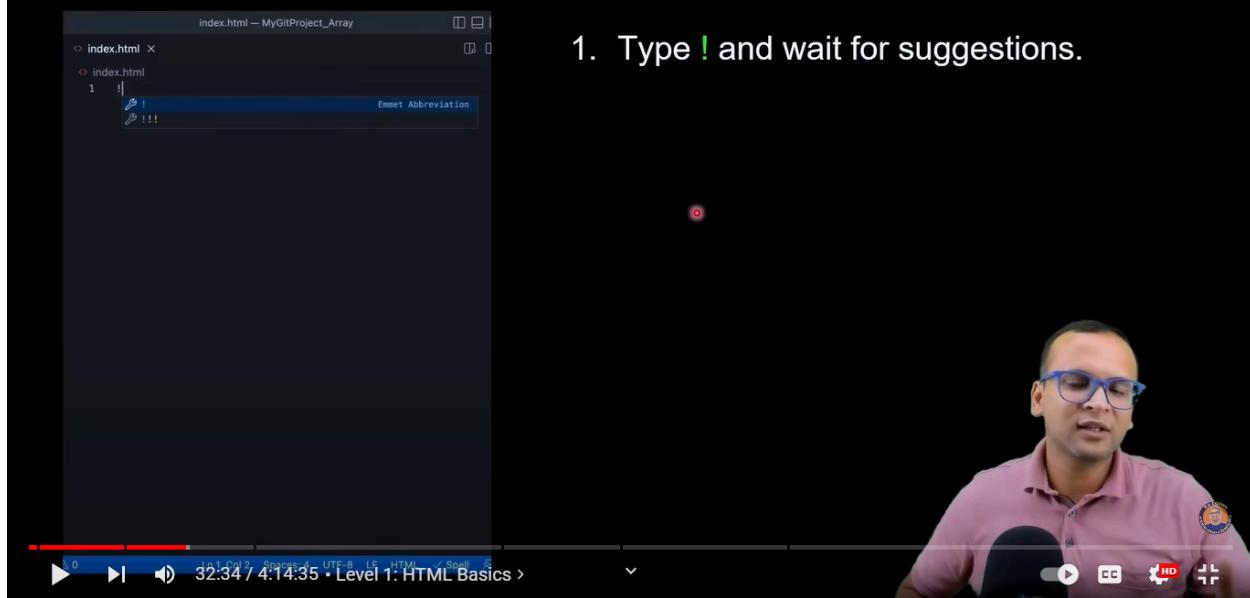
1. Elements that are used to create a website are called HTML Tags.
2. Tags can contain content or other HTML tags.
3. Define elements like text, images, links



# LEVEL 1

## HTML Complete Course (2024) for Beginners | Projects | Notes | Github | Certification

# 2.2 Using Emmet ! To generate code



Using ! it will give suggestion to autofill code, so we dont need to write basic code

Ex

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

## 2.3 Basic HTML Page

```
<!DOCTYPE html>           Defines the HTML Version  
<html lang="en">          Parent of all HTML tags / Root element  
    <head>                  Parent of meta data tags  
        <title>My First Webpage</title>  Title of the web page  
    </head>  
    <body>                  Parent of content tags  
        <h1>Hello World!</h1>  Heading tag  
    </body>  
</html>
```



```
<!DOCTYPE html>  
<html lang="en">      -html main start tag  
<head>      --- in head we can write title and metadata of page  
    <meta charset="UTF-8">      --sample metadata  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Title</title>  
</head>  
<body>      --- from this tag content get start  
    <h1>This is the index file</h1>  ---heading  
</body>  
</html>      -html main end tag
```

LEVEL 1

HTML Complete Course (2024) for Beginners | Projects | Notes | Github | Certification

## 2.4 MDN Documentation



1. Visit <https://developer.mozilla.org/>
2. Official resource for HTML
3. Offers comprehensive guides and tutorials
4. Includes examples for real-world use
5. Updated with latest HTML features
6. Trusted by developers worldwide

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   The body element represents the content of the
7   document.
8   MDN Reference
9 <body>
10
11 </body>
12 </html>
```



|| ▶️ 🔍 37:12 / 4:14:35 • Level 1: HTML Basics > ▾

▶️ CC HD

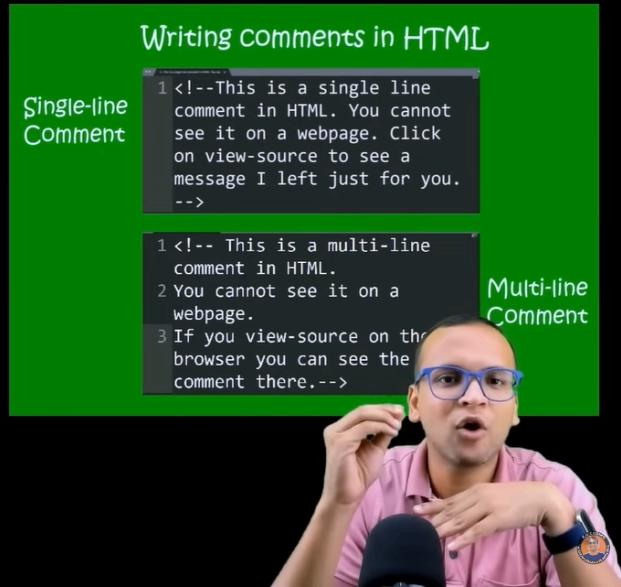
<https://developer.mozilla.org/en-US/docs/Web/HTML>

We can get through cs code, it shows reference

## LEVEL 1

# 2.5 Comments

1. Used to add **notes** in HTML code
2. **Not displayed** on the web page
3. Syntax: `<!-- Comment here -->`
4. Helpful for **code organization**
5. Can be **multi-line** or **single-line**



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Test</title>
</head>
<body>
    <!-- This is sample comment to below header -->
    <!-- This is sample multi line comment
        to below header -->
    <h1>This is for comment</h1>
</body>
</html>
```

LEVEL 1

## 2.6 Case Sensitivity

1. HTML is case-insensitive for tag names
2. Attribute names are also be case-insensitive
3. Best practice: use lowercase for consistency

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

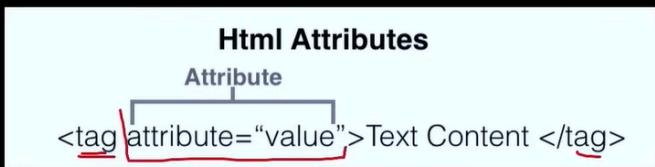


## Project 1

```
<!-- This is basic project created with emmit '!' and adding basic Title  
and head  
    also using comment -->  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>My basic project 1</title>  
</head>  
<body>  
    <h1>Learning basic html code </h1>  
</body>  
</html>
```

# Tags and Attribs

## LEVEL 2 1.1 What are HTML Attributes?



1. Provides additional information about elements
2. Placed **within opening tags**
3. Common examples: href, src, alt
4. Use **name=value** format
5. Can be **single or multiple** per element



## LEVEL 2 1.2 id property

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

This is the header

- 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.



## LEVEL 2

# 2.1 Heading Tag

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>
</body>
</html>
```

The browser window shows the rendered output with the following heading levels:

**Heading 1**  
**Heading 2**  
Heading 3  
Heading 4  
Heading 5  
Heading 6

A man wearing glasses and a pink shirt is visible on the right side of the browser window, speaking into a microphone.

- 1. Defines **headings** in a document
- 2. Ranges from **<h1>** to **<h6>**
- 3. **<h1>** is most important, **<h6>** is least
- 4. Important for **SEO**
- 5. Helps in structuring content

```
<title>My basic project 1</title>
```

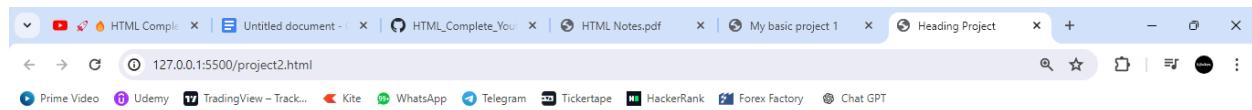
Title is name that shown at browser tab.... As in above img “Learn with KG Coding” at chrome tab name

```
<h1>This is the index file</h1>
```

Head is not related to size of text, it is related to importance of text  
H1,h2,h3,h4... it shows importance for size we use CSS

## Project2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Heading Project</title>
</head>
<body>
    <h1>heading 1</h1>
    <h2>heading 2</h2>
    <h3>heading 3</h3>
    <h4>heading 4</h4>
    <h5>heading 5</h5>
    <h6>heading 6</h6>
</body>
</html>
```



**heading 1**

**heading 2**

**heading 3**

**heading 4**

**heading 5**

**heading 6**



## LEVEL 2

# 2.2 Paragraph Tag

The screenshot shows a video call interface. On the left, a code editor displays an HTML file named 'index.html'. The code includes a title, three paragraphs, and some meta tags. On the right, a browser window shows the rendered output with three distinct paragraphs.

- 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

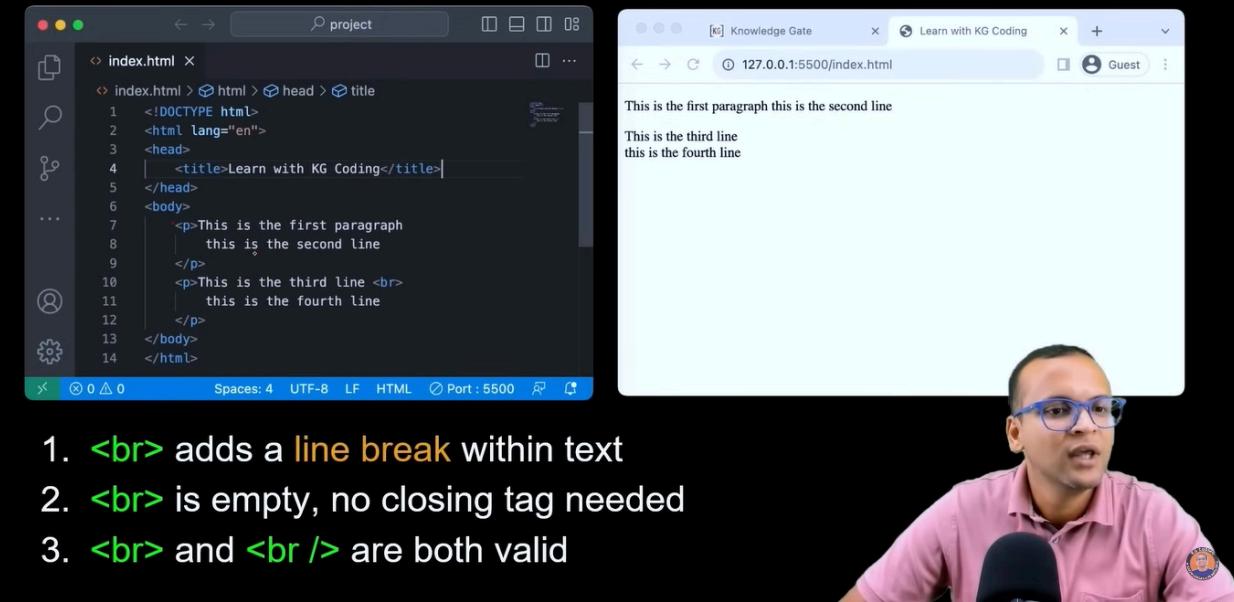


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Paragraph Project</title>
</head>
<body>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
Perspiciatis reprehenderit aut eligendi saepe quia fuga illo, eius officia  
repudiandae, molestias veritatis aliquid ipsa molestiae autem accusamus,  
eum nam odio corrupti.</p>
    <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Sed, ut  
rtwdf wegf teufds! </p>
</body>
</html>
```

In output it gives space in two para

## LEVEL 2

# 2.3 <BR> Tag



This is the first paragraph this is the second line  
This is the third line  
this is the fourth line

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

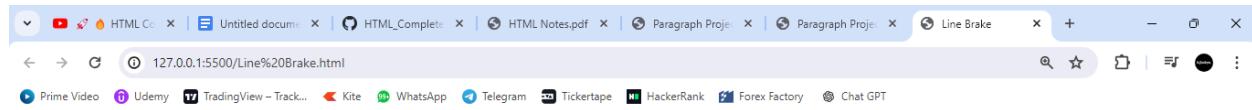


```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Line Brake</title>
</head>
<body>
    <p>Lorem ipsum dolor

        sit amet consectetur adipisicing elit. <br> Sunt, nulla modi! Nisi
        sequi voluptatem tenetur? Voluptates paatur debitis dolor consequatur.
    <br><br><br><br>Incidunt mollitia sunt eaque possimus.</p>
</body>
</html>

```



  sit amet consectetur adipisicing elit.  
Sunt, nulla modi! Nisi sequi voluptatem tenetur? Voluptates paatur debitis dolor consequatur.

Incidunt mollitia sunt eaque possimus.

Here we added lot of space after 'ipsum' bt it took single space, also we added multiple enter after dolor but it does not take that

But as we add single br it took to next line and as we add 4 br it give that 4 enter space

Br don't have closing tag

## LEVEL 2

## 2.3 <HR> Tag

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <h1>Hello World!</h1>
    <hr>
    <p>This is my paragraph</p>
</body>
</html>
```

The browser window shows the rendered output: "Hello World!" followed by a horizontal line (rule) and the text "This is my paragraph". A video player interface is visible on the right side of the browser window.

1. **<hr>** creates a **horizontal rule** or line
2. **<hr>** also empty, acts as a divider

Add horizontal line

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Adding Horizontal Line</title>
</head>
<body>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. <hr> <br>
    Sunt, nulla modi! Nisi sequi voluptatem tenetur? Voluptates paatur debitis
    dolor consequatur. <hr><br><br><br><br>Incidunt mollitia sunt eaque
    possimus.</p>
</body>
</html>
```

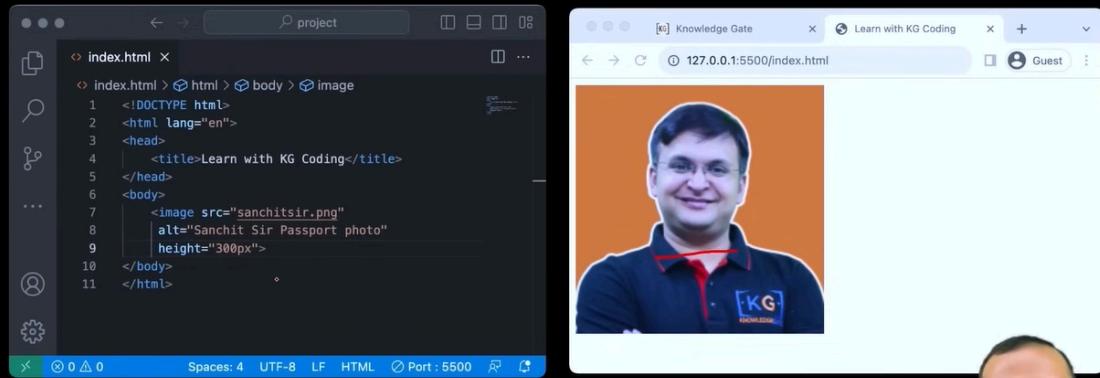
The screenshot shows a browser window displaying the rendered HTML from the previous code block. It features two horizontal lines (rules) separating the text blocks. The browser's address bar shows the URL: 127.0.0.1:5500/Horizontal%20Line.html.

Incidunt mollitia sunt eaque possimus.



## LEVEL 2

# 2.4 Image Tag



The image shows a code editor window on the left containing the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <image src="sanchitsir.png"
        alt="Sanchit Sir Passport photo"
        height="300px">
</body>
</html>
```

To the right is a browser window titled "Learn with KG Coding" showing the rendered image of a man in a blue polo shirt.

1. Used to embed images
2. Utilizes the `src` attribute for image URL
3. `alt` attribute for alternative text
4. Can be resized using `width` and `height`
5. Self-closing, doesn't require an end tag



If image and index.html in same folder

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image</title>
</head>
<body>
    <image src="ajinkya.jpg"
        alt="This is Ajinkya's Image"
        width="300px" />
</body>
</html>
```

Here we can mention width and height

If image in another folder called images

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image</title>
</head>
<body>
    <image src="images/ajinkya.jpg"
        alt="This is Ajinkya's Image"
        width="300px" />
</body>
</html>
```

If image is online also if want to images

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image</title>
</head>
<body>
    <image src="images/ajinkya.jpg"
        alt="This is Ajinkya's Image"
        width="300px" />

    <image src="paste image address here"
        alt="This is online Image"
        width="300px" />
</body>
</html>
```

For video

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Video</title>
</head>
<body>
    <video src="assets/video.mp4"
        alt="This is Ajinkya's Video"
        width="300px"
        autoplay controls loop />

</body>
</html>
```

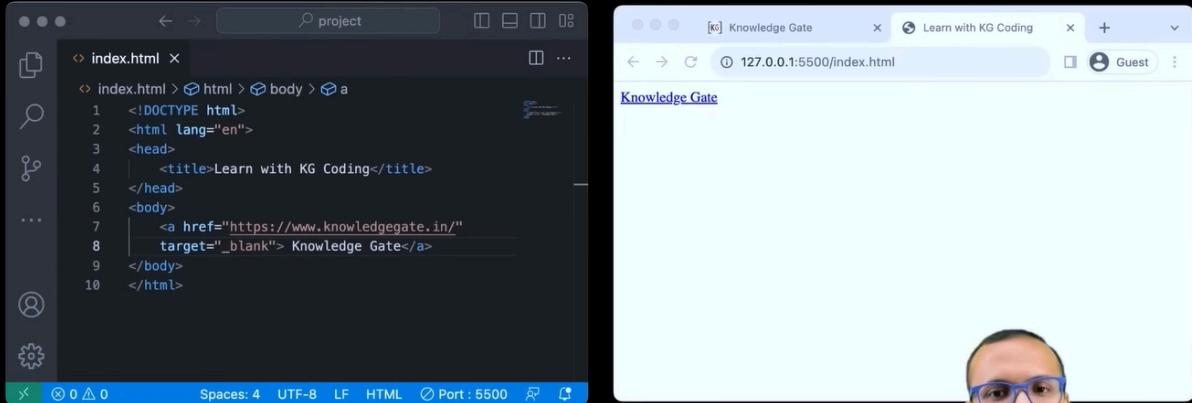
We dont keep videos in image folder, we keep in assets folder

With videos we also keep font files, and other heavy files excluding code files

If video is not playing automatically then we can add “muted” and try it

## LEVEL 2

# 2.6 Anchor Tag



The image shows a code editor on the left and a browser window on the right. The code editor displays the file 'index.html' with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <a href="https://www.knowledgegate.in/" target="_blank"> Knowledge Gate</a>
</body>
</html>
```

The browser window shows the rendered HTML with a single link labeled 'Knowledge Gate'.

1. Used for creating **hyperlinks**
2. Requires **href** attribute for URL
3. Can link to external sites or internal pages
4. Supports **target** attribute to control link behavior

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Anchor</title>
</head>
<body>
    <a href="https://www.instagram.com/"> Click for Instagram</a><br>
    <a href="https://www.google.com/"> Click for Google</a><br>
    <a href="https://www.facebook.com/"> Click for Facebook</a>
</body>
</html>
```

## LEVEL 2 Bold/Italic/Underline/Strikethrough

The screenshot displays a developer's environment. On the left is a code editor showing the file 'index.html' with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <b>Bold</b> <hr>
    <i>Italic</i> <hr>
    <u>Underline</u> <hr>
    <s>Strikethrough</s>
</body>
</html>
```

On the right is a browser window titled 'Knowledge Gate' showing the rendered output:

**Bold**  
*Italic*  
Underline  
~~Strikethrough~~

A video overlay of a man in a pink shirt and glasses speaking into a microphone is visible on the right.

1. **<b>** makes text **bold**
2. **<i>** makes text **italic**
3. **<u>** underlines **text**
4. **<s>** or **<strike>** applies **strikethrough**
5. Primarily used for text styling and emphasis

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Font Style</title>
</head>
<body>
    <b>This line is in Bold</b><hr>
    <i>This line is in Italic</i><hr>
    <u>This line is with Underline</u><hr>
    <s>This line is with Strikethrough</s><hr>
</body>
</html>
```

Hr is used for drawing horizontal line between them

## LEVEL 2

### 2.8 Pre Tag

The code editor shows the file `index.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    first line
    more spaces
<pre>
    first line
    more spaces
</pre>
</body>
</html>
```

The browser preview shows the rendered output with the text "first line" and "more spaces" on the first line, followed by the text "first line" and "more spaces" on a new line, demonstrating the preservation of whitespace.

1. Preserves text formatting
2. Maintains whitespace and line breaks
3. Useful for displaying code
4. Enclosed within `<pre>` and `</pre>` tags

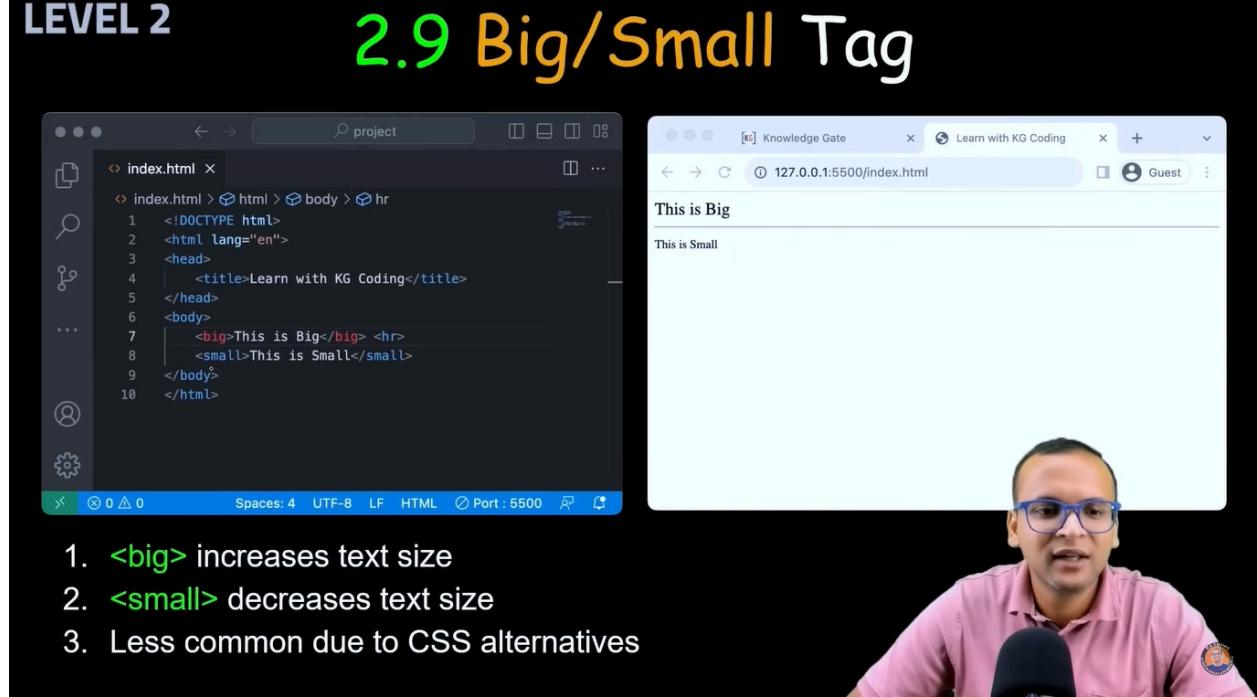
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pre Tag</title>
</head>
<body>
    this      website      website
    hrhsa      hefw

    wfhubwuhwf
    <pre>the is      sfssd
        urtto      otis

        hskashe </pre>
</body>
</html>
```

## LEVEL 2

# 2.9 Big/Small Tag



The screenshot shows a video call interface. On the right, a person wearing a pink polo shirt and blue-rimmed glasses is speaking into a black microphone. On the left, there is a code editor window titled "index.html" and a browser window titled "Knowledge Gate". The code editor shows the following HTML:

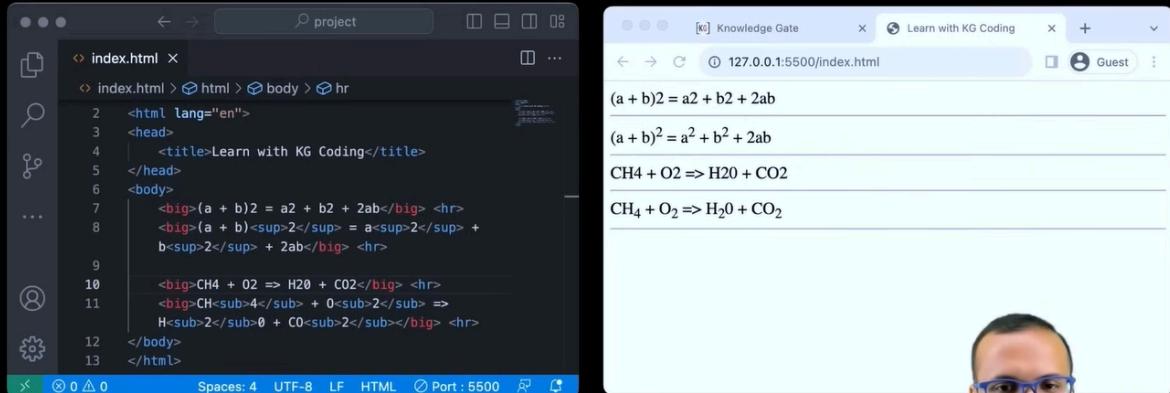
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <big>This is Big</big> <br>
    <small>This is Small</small>
</body>
</html>
```

The browser window displays the rendered HTML with the text "This is Big" in a large font and "This is Small" in a smaller font, separated by a line break.

1. **<big>** increases text size
2. **<small>** decreases text size
3. Less common due to CSS alternatives

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Big Small</title>
</head>
<body>
    <big>Hello im big</big><br>
    <small>Hello im small</small>
</body>
</html>
```

## LEVEL 2.10 Superscript/Subscript Tag



The image shows a code editor window on the left and a browser window on the right. The code editor displays the following HTML code:

```
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <big>(a + b)2 = a2 + b2 + 2ab</big> <br>
    <big>(a + b)2 = a2 + b2 + 2ab</big> <br>
    <big>CH4 + O2 => H2O + CO2</big> <br>
    <big>CH4 + O2 => H2O + CO2</big> <br>
</body>
</html>
```

The browser window shows the rendered output of the code, displaying mathematical equations and chemical reactions correctly formatted with superscripts and subscripts.

1. `<sup>` makes text superscript
2. `<sub>` makes text subscript
3. Used for mathematical equations, footnotes
4. Does not change font size, just position



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sup Sub</title>
</head>
<body>
    (a+b)2= a2+b2+2ab <br><br>
    (a+b)2=a2+b2+2ab <br><br>
    (a+b)2=a2+b2+2ab
</body>
</html>
```

`<sup>` superscript  
`<sub>` subscript

## LEVEL<sup>2</sup> 5.1 Character Entity Reference

1. Used to display **reserved** or **special** characters
2. Syntax often starts with & and ends with ; (e.g., &amp; for &)

&nbsp;	-	&ndash;	-	&minus;	°	&deg;	Δ	&Delta;	α	&ahar;	
€	&euro;	—	&mdash;	±	&plusmn;	°	&ordm;	Λ	&Lambda;	β	&beta;
ƒ	&cent;	…	&hellip;	√	&radic;	ª	&ordf;	Θ	&Theta;	γ	&gamma;
£	&pound;	§	&sect;	∞	&infin;	¹	&sup1;	Ξ	&Xi;	δ	&delta;
¥	&yen;	¶	&para;	∞	&prop;	²	&sup2;	Π	&Pi;	ε	&epsilon;
¤	&curren;	†	&dagger;	×	&times;	³	&sup3;	Σ	&Sigma;	ζ	&zeta;
ƒ	&fnof;	‡	&Dagger;	÷	&divide;	¼	&frac14;	Φ	&Phi;	η	&eta;
©	&copy;		&iexcl;	~	&sim;	½	&frac12;	Ψ	&Psi;	θ	&theta;
®	&reg;	¿	&iquest;	≈	&asymp;	¾	&frac34;	Ω	&Omega;	ι	&ic;
™	&trade;	%	&permil;	≡	&cong;	⋮	&there4;	∇	&nabla;	κ	



## LEVEL 2

# Project Level 2

Must-Use HTML Tags

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** and **strike through** in one line.
4. Write third equation of motion using **superscript** and **subscript**.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Project 2</title>
</head>
<body>
    <a href="https://www.fb.com" target="_blank">
              </a><br></a>
        <hr>

    <p>This is paragraph line </p><br><hr>
    <h1>This is heading h1</h1><br>
    <h4>This is heading h4 </h4><hr>

    <b>My</b> <i>name</i> <u>is</u> <s>Varesh Patil</s><hr>

    C12H22011 <br>
    C<sub>12</sub>H<sub>22</sub>O<sub>11</sub> <br>
    C<sup>12</sup>H<sup>22</sup>O<sup>11</sup>

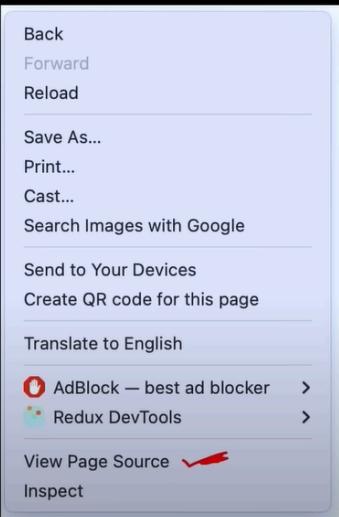
</body>
</html>
```

LEVEL 3

HTML Complete Course (2024) for Beginners | Projects | Notes | Github | Certification



## 1.1 View Page Source

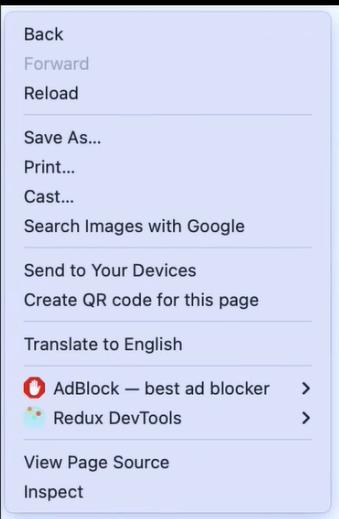


1. Displays raw HTML and CSS
2. Useful for debugging and learning ✓
3. Shows external files like JavaScript links ✓



LEVEL 3

## 1.2 Inspect Element



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 element on page



LEVEL 3

# Level 3

Browser Tools



## 2. Responsive Design



LEVEL 3

### 2.1 Different Screen Sizes



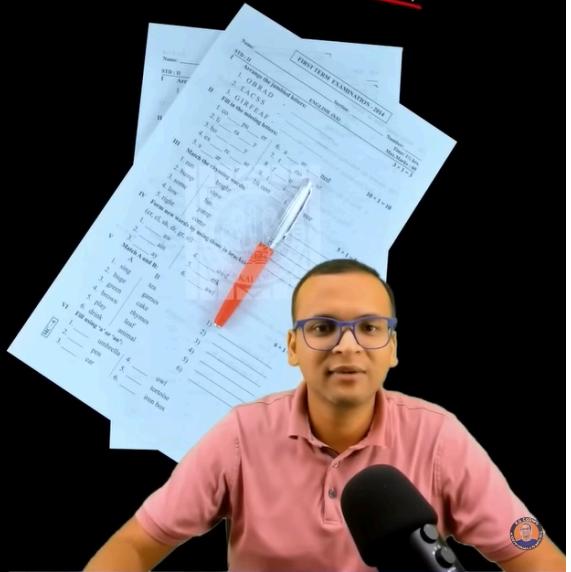
1. Adapts layout for different screen sizes
2. Flexible layouts
3. Optimizes images and assets
4. Enhances user experience on mobile and desktop



## LEVEL 3.4 Changes happening at Client

1. Changes made are temporary
2. Affect only the current session
3. Not saved to the server
4. Reset upon page reload
5. Useful for testing, not permanent fixes

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



## LEVEL 3 4.1 Using validator.w3.org

A screenshot of the Nu Html Checker web application. The interface has a blue header bar with the text 'Nu Html Checker'. Below it, a message says: 'This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change'. Underneath, it says 'Showing results for contents of text-input area'. There's a 'Checker Input' section with a text area containing some HTML code. A 'Check' button is below the text area. At the bottom, there's a 'Message Filtering' section with a message: 'Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.' Below that is a list of errors: '1. Error Stray start tag &lt;html&gt;. From line 9, column 1 to line 9, column 6' with a red checkmark icon next to it. The footer of the browser window shows various icons.

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



## LEVEL 3

# Project Level 3

To exit full screen, press Esc

Browser Tools

1. Save Source of Instagram in a file and check the render.
2. Inspect the likes element on the page and read the code to understand.
3. Change number of likes on Your Instagram post
4. Validate the page we created in last project. ✓



## LEVEL 4

# Level 4

## HTML and Project Structure

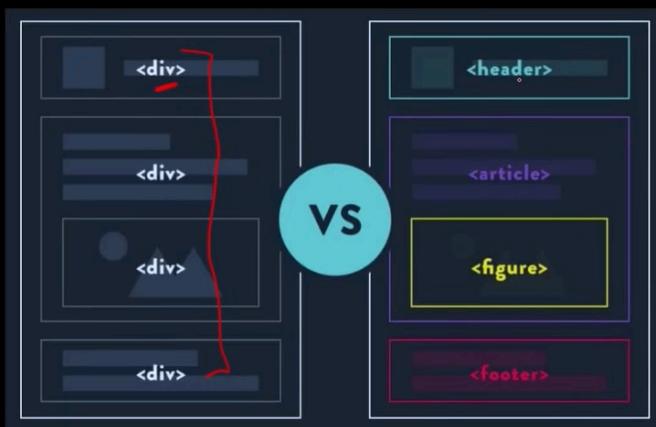
1. Semantic Tags
  1. Semantic / Non-Semantic Tags
2. Body Tags
  1. Header Tag
  2. Main Tag
    1. Section Tag
    2. Article Tag
    3. Aside Tag
  3. Footer Tag
3. Folder Structure
  1. Recommended Folder structure
4. More Tags
  1. Navigation tags
  2. Block / Inline Elements
  3. Div tags
  4. Span Tags



## LEVEL 4

# Level 4

## HTML and Project Structure



LEVEL 4

## 1.1 Semantic/Non-Semantic Tags

### Semantic Tags

- Meaningful: Describe content.
- SEO: Good for search engines.
- Accessibility: Useful for screen readers.
- Examples: `<header>`, `<footer>`, `<article>`, `<section>`, `<nav>`.

### Non-Semantic Tags

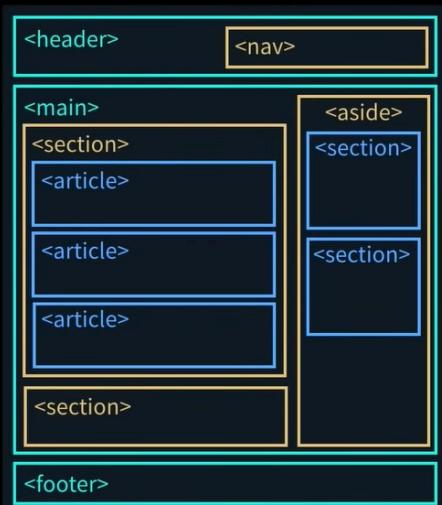
- Generic: No specific meaning.
- For Styling: Used for layout.
- No SEO: Not SEO-friendly.
- Examples: `<div>`, `<span>`, `<i>`, `<b>`.



LEVEL 4

## Level 4

### HTML and Project Structure

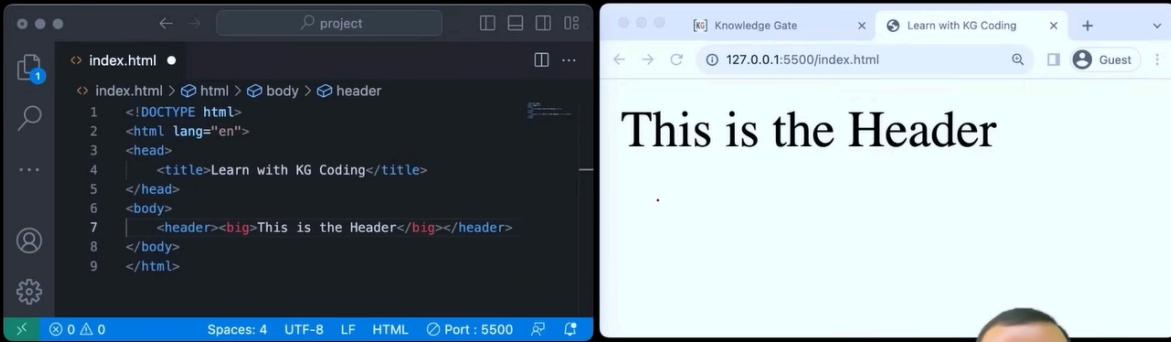


## 2. Body Tags



## LEVEL 4

# 2.1 Header Tag



The image shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <header><big>This is the Header</big></header>
</body>
</html>
```

The browser window shows the rendered output: "This is the Header".

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 be used within `<article>` or `<section>` tags.
4. Multiple Instances: Can be used more than once on a page without causing conflicts.

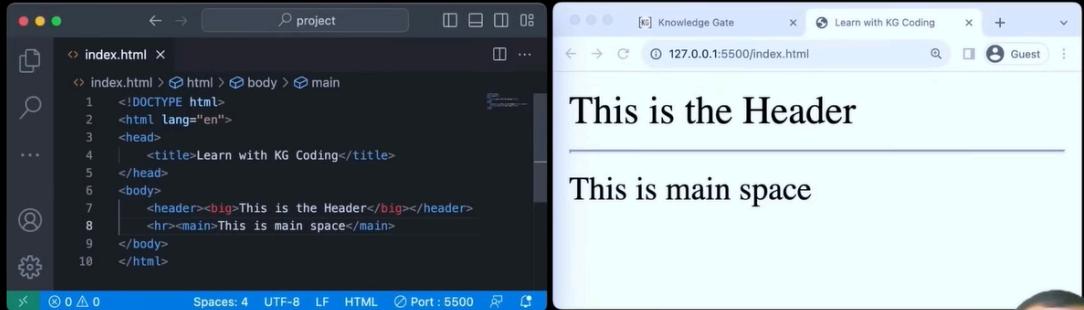


Its good practice to use `<big>`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Header</title>
</head>
<body>
    <header><big>This is header</big></header>
</body>
</html>
```

## LEVEL 4

# 2.2 Main Tag



The image shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <header><b>This is the Header</b></header>
    <hr><main>This is main space</main>
</body>
</html>
```

The browser window shows the rendered output:

This is the Header

---

This is main space

1. **Purpose:** Encloses the primary content of a webpage.
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 Sidebars:** Excludes content repeated across multiple pages, navigation or footer.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>main</title>
</head>
<body>
    <main>This is main tag</main>
</body>
</html>
```

## LEVEL 4

### 2.2.1 Section Tag

The screenshot shows a video call interface. On the left, a code editor window titled "index.html" displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <main>
        <section>Technologies: HTML/CSS</section>
        <section>Courses: OS/DBMS</section>
        <section>Hobbies: Eating</section>
    </main>
</body>
</html>
```

On the right, a browser window titled "Knowledge Gate" shows the rendered output:

Technologies: HTML/CSS  
Courses: OS/DBMS  
Hobbies: Eating

A man with glasses and a mustache, wearing a black t-shirt with the text "I need" partially visible, is speaking into a microphone.

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 hierarchy.
4. Nested: Can be nested within other `<section>` or `<article>` tags.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Section Tag</title>
</head>
<body>
    <main>
        <section>Section 1</section>
        <section>sec 2</section>
        <section>sec 3</section>
    </main>
</body>
</html>
```

## LEVEL 4

### 2.2.3 Article Tag

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Article</title>
</head>
<body>
    <main>
        <article>
            KnowledgeGate YouTube channel is a popular educational resource focused on computer science and engineering topics. It offers tutorials, interview tips, and easy-to-understand explanations, making it a go-to channel for students and professionals alike.
        </article>
    </main>
</body>
</html>
```

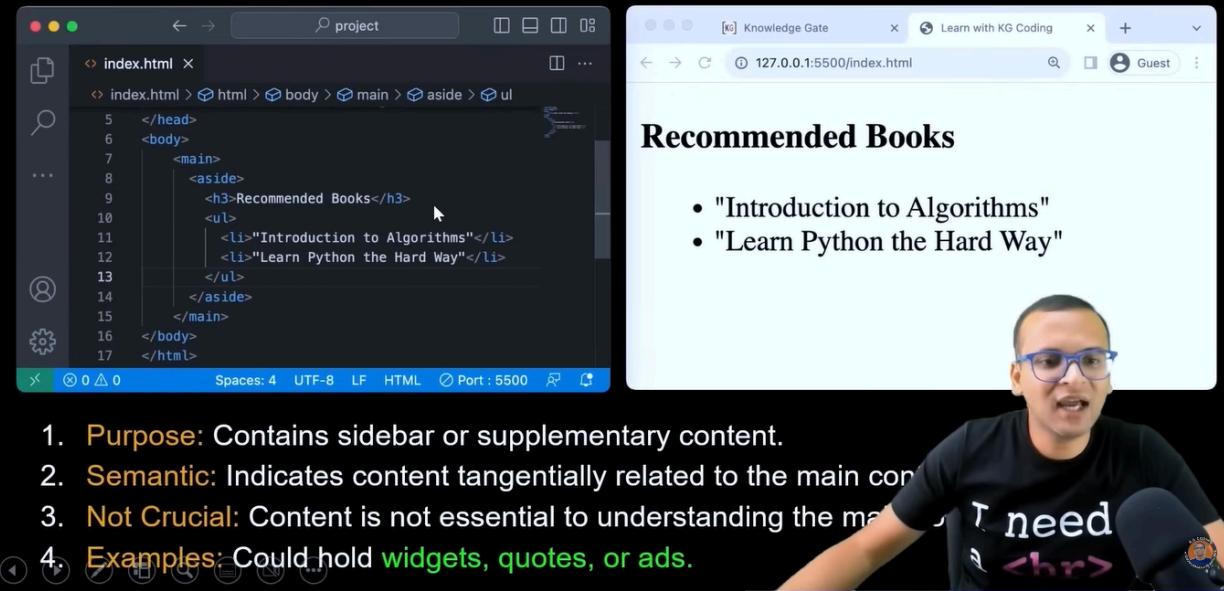
The browser window shows the rendered content of the HTML, which is a single paragraph of text: "KnowledgeGate YouTube channel is a popular educational resource focused on computer science and engineering topics. It offers tutorials, interview tips, and easy-to-understand explanations, making it a go-to channel for students and professionals alike." Below the browser window, there is a video frame of a man wearing glasses and a t-shirt that says "I need a <br>".

- Purpose:** Encloses content that stands alone, like a [blog post](#) or [news article](#).
- Semantic:** Provides contextual meaning.
- Independence:** Content should make sense even if taken out of its context.
- Multiple Instances:** Can be used multiple times on the same page.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Article</title>
</head>
<body>
    <main>
        <article>
            Lorem, ipsum dolor sit amet consectetur adipisicing elit. Eos modi maiores accusantium doloribus officia a cum blanditiis nam amet laudantium?
        </article>
    </main>
</body>
</html>
```

## LEVEL 4

### 2.2.3 Aside Tag



The image shows a video call interface. On the left, a developer's screen is visible, displaying a code editor with an open file named 'index.html'. The code editor shows the following HTML structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Aside Tag</title>
</head>
<body>
    <main>
        <aside>
            this is aside tag <br>
            gsjhds <br>
            hdhd
        </aside>
    </main>
</body>
</html>
```

To the right of the developer's screen is a browser window titled 'Knowledge Gate' showing the rendered HTML. The browser displays the heading 'Recommended Books' and a list of books:

- "Introduction to Algorithms"
- "Learn Python the Hard Way"

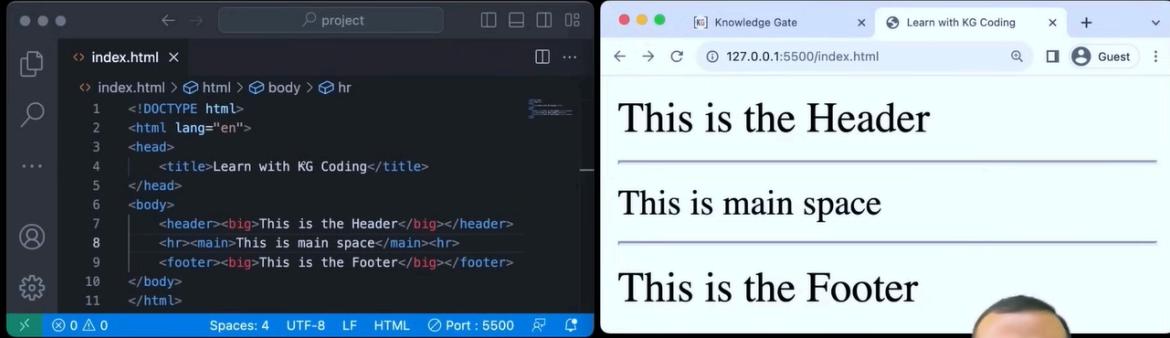
A video feed of a person wearing glasses and a black t-shirt is overlaid on the browser window.

1. **Purpose:** Contains sidebar or supplementary content.
2. **Semantic:** Indicates content tangentially related to the main content.
3. **Not Crucial:** Content is not essential to understanding the main content.
4. **Examples:** Could hold **widgets, quotes, or ads**.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Aside Tag</title>
</head>
<body>
    <main>
        <aside>
            this is aside tag <br>
            gsjhds <br>
            hdhd
        </aside>
    </main>
</body>
</html>
```

## LEVEL 4

# 2.3 Footer Tag



The image shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <header><b>This is the Header</b></header>
    <hr><main>This is main space</main><hr>
    <footer><b>This is the Footer</b></footer>
</body>
</html>
```

The browser window shows the rendered HTML with the title "Learn with KG Coding". It contains three horizontal sections: "This is the Header" at the top, "This is main space" in the middle, and "This is the Footer" at the bottom.

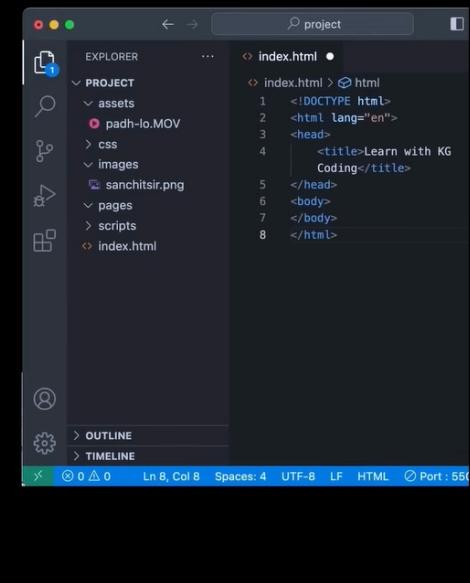
1. Purpose: For footer content like extra info or links.
2. Semantic: Provides meaning to enclosed content.
3. Location: Typically at the **bottom** of pages or sections.
4. Content: Includes copyrights, contact info, and social links.
5. Multiple Instances: Can be used more than once on a page.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Footer</title>
</head>
<body>
    <Footer>
        this is footer
    </Footer>
</body>
</html>
```

## LEVEL 4

# 3.1 Recommended Folder Structure

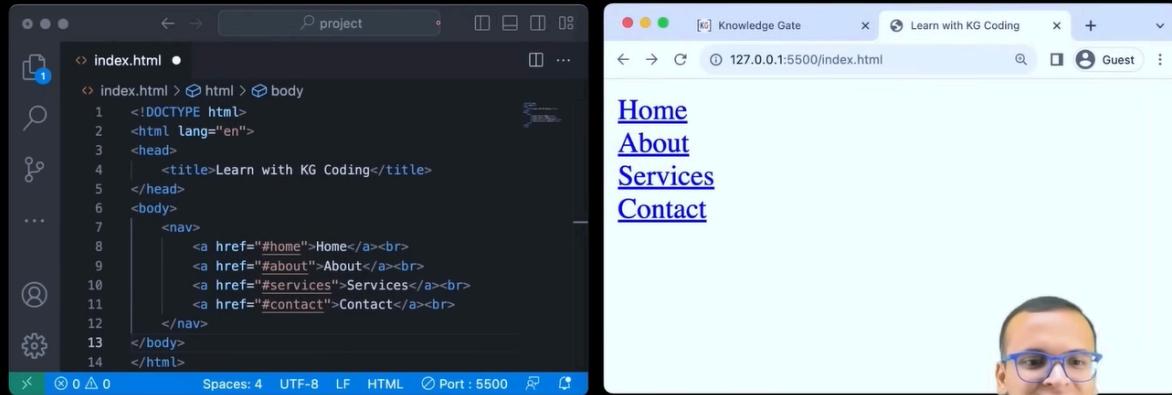


1. **Root Directory:** Main folder containing all website files.
2. **HTML Files:** Store main .html files at the root level for easy access.
3. **CSS Folder:** Create a css/ folder for all Cascading Style Sheets.
4. **JS Folder:** Use a scripts/ folder for JavaScript files.
5. **Images Folder:** Store 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-folders to categorize content.



## LEVEL 4

# 4.1 Navigation Tags



The image shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <nav>
        <a href="#home">Home</a><br>
        <a href="#about">About</a><br>
        <a href="#services">Services</a><br>
        <a href="#contact">Contact</a><br>
    </nav>
</body>
</html>
```

The browser window shows the rendered output: [Home](#), [About](#), [Services](#), and [Contact](#).

1. **Purpose:** Encloses navigation links or menus.
2. **Semantic:** Signals that the content is meant for navigating the site.
3. **Common Content:** Usually contains lists `<ul>`, `<ol>` of links `<a>`.
4. **Accessibility:** Aids screen readers in identifying site navigation



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Navigation</title>
</head>
<body>
    <a href="#Contact">Contact</a>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Eos necessitatibus porro non eum dolorem ex! Doloremque numquam rem non sed id animi quae excepturi repellendus deleniti repellat. Optio, iste deleniti.</p>
    <p id="Contact"> Contact: +91 9158147547</p>
</body>
</html>
```

There contact button will get created, and if we click on contact then it will redirect or navigate to below mentioned contact no.

## LEVEL 4 4.2 Block / Inline Elements

### Block Elements

- New Line: Start on a new line.
- Full Width: Take up all horizontal space.
- Styling: Can have margins and padding.
- Size: Width and height can be set.
- Examples: `<div>`, `<p>`, `<h1>`, `<ul>`, `<li>`.

### Inline Elements

- Flow: Stay in line with text.
- Width: Just as wide as the content.
- No Break: No new line between elements.
- Limited Styling: Can't style easily.
- Examples: `<span>`, `<small>`, `<em>`, `<img>`.



In block element it will consume all horizontal space even if it is blank and it will take next sentence to next line

Ex

I'm ajinkya patil

Im from Islampur

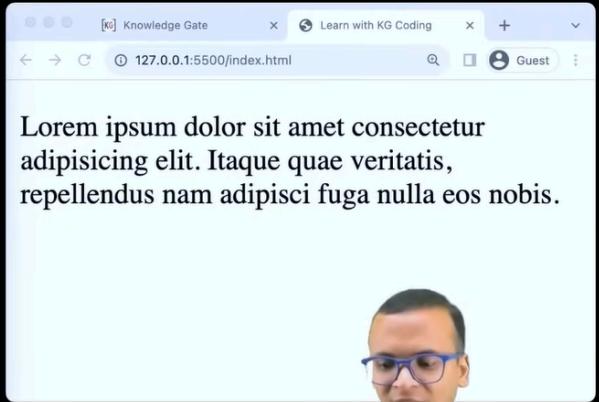
In inline it does not consume blank space, if we add another line with inline element then it will and on same line, will not go to next line

Ex

I'm ajinkya patil Im from Islampur

## LEVEL 4

# 4.3 Div Tags



A screenshot of a web browser window titled "Knowledge Gate" showing the URL "127.0.0.1:5500/index.html". The page content is a single paragraph of placeholder text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque quae veritatis, repellendus nam adipisci fuga nulla eos nobis."

On the left, there is a code editor window titled "index.html" showing the corresponding HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Div tag</title>
</head>
<body>
    <div>
        <p>Im Ajinkya patil</p>
        <h1>Im from urun islampur</h1>
    </div>
</body>
</html>
```

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.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Div tag</title>
</head>
<body>
    <div>
        <p>Im Ajinkya patil</p>
        <h1>Im from urun islampur</h1>
    </div>
</body>
</html>
```

When u put content in div then you can add same styling and all other things on div and it will apply on all grp, don't need to apply individually

## LEVEL 4

# 4.4 Span Tags

This is a highlighted word in a paragraph.

1. **Purpose:** Used for inline elements to style or manipulate a portion of text.
2. **Non-Semantic:** Doesn't add specific meaning to the enclosed text.
3. **Styling:** Commonly used for changing color, font, or adding effects via CSS.
4. **Inline Nature:** Doesn't break text flow or create a new block-level element.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Span tag</title>
</head>
<body>
    <div>
        <p>I'm Ajinkya patil, I'm from <span>urun islampur</span></p>
    </div>
</body>
</html>
```

Without span it will consider it as one line due to div grouping

But due to span it will break urun ismapur separate so we can do many css changes on it, like color or highlight

## LEVEL 4

# Project Level 4

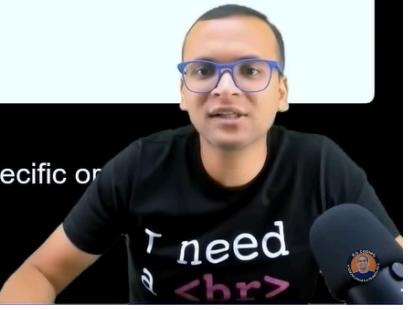
HTML and Project Structure

1. Create a **page** with header, footer, main(section, article, aside tag).
2. Make sure the project from level 3 has correct **folder structure**.
3. Create **groupings** of multiple tags using div.
4. Create **navigation** to important sections of your page.



## LEVEL 5

# 1.1 Ordered Lists



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<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>
```

The browser window shows the rendered output of the code:

1. Wake up
2. Brush teeth
3. Take a shower
4. Have breakfast
5. Go to work

1. Purpose: Used for creating lists with items that have a specific order.
2. Default: Items are automatically numbered.
3. Nesting: Can be nested within other lists.

## LEVEL 5

# 1.2 Types of Ordered Lists

### Ordered Lists

- 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"`

A. Apple  
B. Banana  
C. Cherry  
D. Dragonfruit

a. Apple  
b. Banana  
c. Cherry  
d. Dragonfruit

I. Apple  
II. Banana  
III. Cherry  
IV. Dragonfruit

i. Apple  
ii. Banana  
iii. Cherry  
iv. Dragonfruit

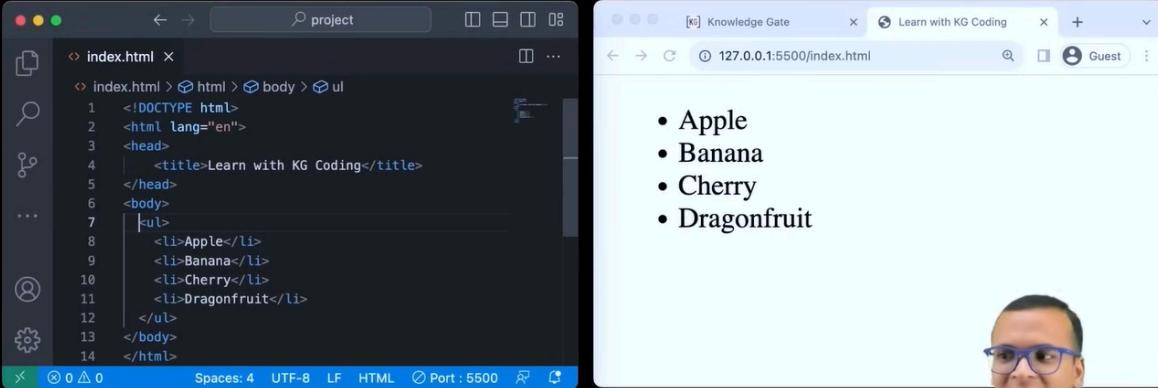


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Order list</title>
</head>
<body>
    <ol>
        <li>ABC</li>
        <li>BCD</li>
    </ol>
    <ol type="a">
        <li>Patil</li>
        <ol type="i">
            <li>Ankush</li>
            <li>Ajinkyaa</li>
            <li>Amit</li>
        </ol>
        <li>yadav</li>
        <li>jadhav</li>
    </ol>
</body>
</html>
```

By default it takes 1,2,3,4,....

## LEVEL 5

# 1.3 Unordered Lists



The image shows a code editor window on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <ul>
        <li>Apple</li>
        <li>Banana</li>
        <li>Cherry</li>
        <li>Dragonfruit</li>
    </ul>
</body>
</html>
```

The browser window shows the rendered output of the code, which is a bulleted list of four items: Apple, Banana, Cherry, and Dragonfruit.

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.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>unOrder list</title>
</head>
<body>
    <ul>
        <li>ABC</li>
        <li>BCD</li>
    </ul>
    <ul type="a">
        <li>Patil</li>
        <ol type="i">
            <li>Ankush</li>
            <li>Ajinkya</li>
        </ol>
        <li>yadav</li>
        <li>jadhav</li>
    </ul>
</body>
</html>
```

## LEVEL 5

# 2.1 <tr>, <td>, <th> Tags

The screenshot shows a video call interface. On the right, a person wearing glasses and a black t-shirt with text on it is speaking. On the left, there is a code editor window titled "index.html" and a browser window titled "Knowledge Gate". The browser shows a simple HTML table with three rows and three columns. The first row contains header cells ("Name", "Age", "Email"). The second row contains data cells ("John", "30", "john@email.com"). The third row contains data cells ("Jane", "25", "jane@email.com").

```
<tr>
<th>Name</th>
<th>Age</th>
<th>Email</th>
</tr>
<tr>
<td>John</td>
<td>30</td>
<td>john@email.com</td>
</tr>
```

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 centered by default.
3. **<td> Table Data** : This Holds the actual data.

## LEVEL 5

# 2.2 Captions

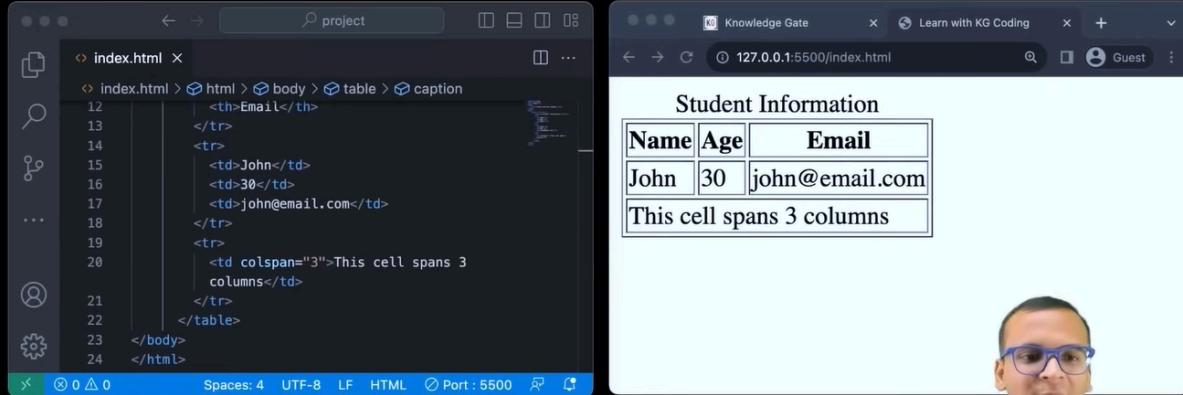
The screenshot shows a video call interface. On the right, a person wearing glasses and a black t-shirt with text on it is speaking. On the left, there is a code editor window titled "index.html" and a browser window titled "Knowledge Gate". The browser shows a table with a caption "Student Information" above it. The table has three rows and three columns. The first row contains header cells ("Name", "Age", "Email"). The second row contains data cells ("John", "30", "john@email.com"). The third row contains data cells ("Jane", "25", "jane@email.com").

```
<caption>Student Information</caption>
<tr>
<th>Name</th>
<th>Age</th>
<th>Email</th>
</tr>
<tr>
<td>John</td>
<td>30</td>
<td>john@email.com</td>
</tr>
```

1. **Purpose**: Provides a title or description for a table.
2. **Placement**: Must be inserted **immediately after** the **<table>** opening tag.
3. **Alignment**: **Centered** above the table by default.
4. **Accessibility**: Helps screen readers understand the table's purpose.

## LEVEL 5

# 2.3 Col Spans



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Table</title>
</head>
<body>
    <table border="1">
        <caption>Family data</caption>
        <thead>
            <tr><th>Sr</th>
            <th>Name</th>
            <th>Age</th>
        </tr>
    </thead>
    <tbody>
        <tr><td>1</td><td>Ajinkya</td><td>26</td></tr>
        <tr><td>2</td><td>Varesh</td><td>16</td></tr>
        <tr><td>3</td><td>Viren</td><td>7</td></tr>
        <tr><td colspan="2">taking 2 column span</td><td>28</td></tr>
    </tbody>
</table>
</body>
</html>
```

The browser window shows a table with three columns: Name, Age, and Email. The last row contains a single cell with the text "This cell spans 3 columns".

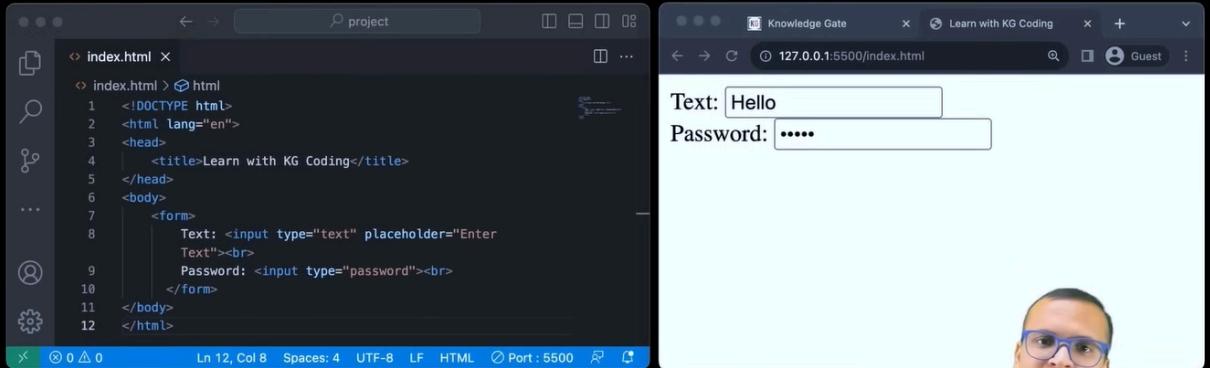
1. Attribute: Uses the colspan attribute in `<td>` or `<th>` tags.
2. Purpose: Allows a cell to span multiple columns horizontally.
3. Alignment: Takes the space of the specified number of columns.
4. Layout: Useful for combining cells to create complex table layouts.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Table</title>
</head>
<body>
    <table border="1">
        <caption>Family data</caption>
        <thead>
            <tr><th>Sr</th>
            <th>Name</th>
            <th>Age</th>
        </tr>
    </thead>
    <tbody>
        <tr><td>1</td><td>Ajinkya</td><td>26</td></tr>
        <tr><td>2</td><td>Varesh</td><td>16</td></tr>
        <tr><td>3</td><td>Viren</td><td>7</td></tr>
        <tr><td colspan="2">taking 2 column span</td><td>28</td></tr>
    </tbody>
</table>
</body>
</html>
```

## LEVEL 5

# 3.1 Input Tag



The image shows a code editor window on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form>
        Text: <input type="text" placeholder="Enter Text"><br>
        Password: <input type="password"><br>
    </form>
</body>
</html>
```

The browser window shows the rendered HTML with two input fields. The first input field is a text input with the placeholder "Enter Text" and the value "Hello". The second input field is a password input with the placeholder "Enter Password" and the value "\*\*\*\*\*".

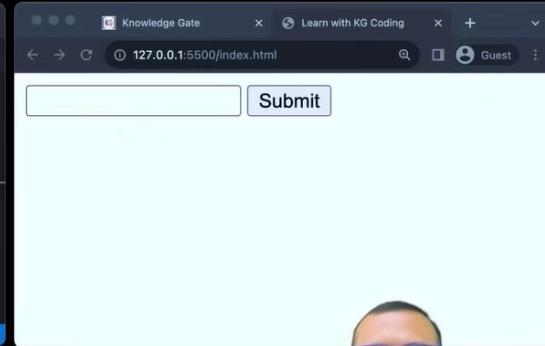
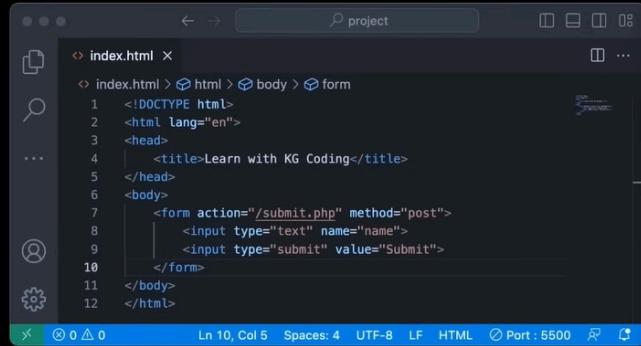
1. **Purpose:** Used within a `<form>` element to collect user input.
2. **Self-Closing:** The `<input>` tag is self-closing; doesn't require a closing tag.
3. **Attributes:** Common attributes are `name`, `value`, `placeholder`, and `required`.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Tag</title>
</head>
<body>
    <form>
        Username: <input type="text" placeholder="Enter Username"
required="True"> <br><br>
        Password: <input type="password" placeholder="Enter Password">
    </form>
</body>
</html>
```

## LEVEL 5

# 3.2 Action attribute



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.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Tag</title>
</head>
<body>
    <form action="/login-data">
        Username: <input type="text" placeholder="Enter Username" required="True"> <br><br>
        Password: <input type="password" placeholder="Enter Password">
    <br>
        <input type="submit">
    </form>
</body>
</html>
```

## LEVEL 5

# 3.3 Name and Value property

The screenshot shows a developer's environment with two windows. On the left is a code editor window titled 'index.html' containing the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form action="/submit.php" method="post">
        <input type="text" name="username" value="John">
    </form>
</body>
</html>
```

On the right is a browser window titled 'Knowledge Gate' showing the URL '127.0.0.1:5500/index.html'. Inside the browser, there is a single text input field containing the text 'John'.

**• `name` Property:**

- **ID for Data:** Identifies form elements when submitting.
- **Unique:** Should be unique to each element for clarity.

**• `value` Property:**

- **Default Data:** Sets initial value for input elements.
- **Sent to Server:** This is the data sent when form is submitted.

A person wearing glasses and a t-shirt with text on it is visible in the background on the right side of the screen.

## LEVEL 5

## 3.4 Label Tag

The screenshot shows a video conference interface. On the left, a code editor displays the file 'index.html' with the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form action="/submit.php" method="post">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username">
    </form>
</body>
</html>
```

On the right, a browser window shows the rendered HTML with a red arrow pointing to the 'Username:' label. Below the video, a man wearing glasses and a t-shirt with text on it is speaking.

- Purpose: Adds a text description to form elements.
- for Attribute: Connects the label to a specific form element using the element's id.
- Accessibility: Makes the form more accessible.
- Readability: Enhances form readability and usability.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Tag</title>
</head>
<body>
    <form action="/login-data">
        <label for="user-id">Username: </label>
        <input type="text" id="user-id" placeholder="Enter Username"
required="True"> <br><br>

        <label for="password">Password: </label>
        <input type="password" id="password" placeholder="Enter Password">
<br>
        <input type="submit">
    </form>
</body>
</html>
```

## LEVEL 5

# 3.5 Input type: Date

The image shows a developer's environment. On the left is a code editor with the file 'index.html' open. The code defines a form with three input fields: a text input ('Text'), a password input ('Password'), and a date input ('Date'). The date input has a placeholder 'Enter Date'. On the right is a browser window displaying the form. The 'Text' field contains 'hello', the 'Password' field contains '\*\*\*\*\*', and the 'Date' field contains '16/08/2023'. Below the browser is a video feed of a person wearing glasses and a t-shirt that says 'I need <br>'.

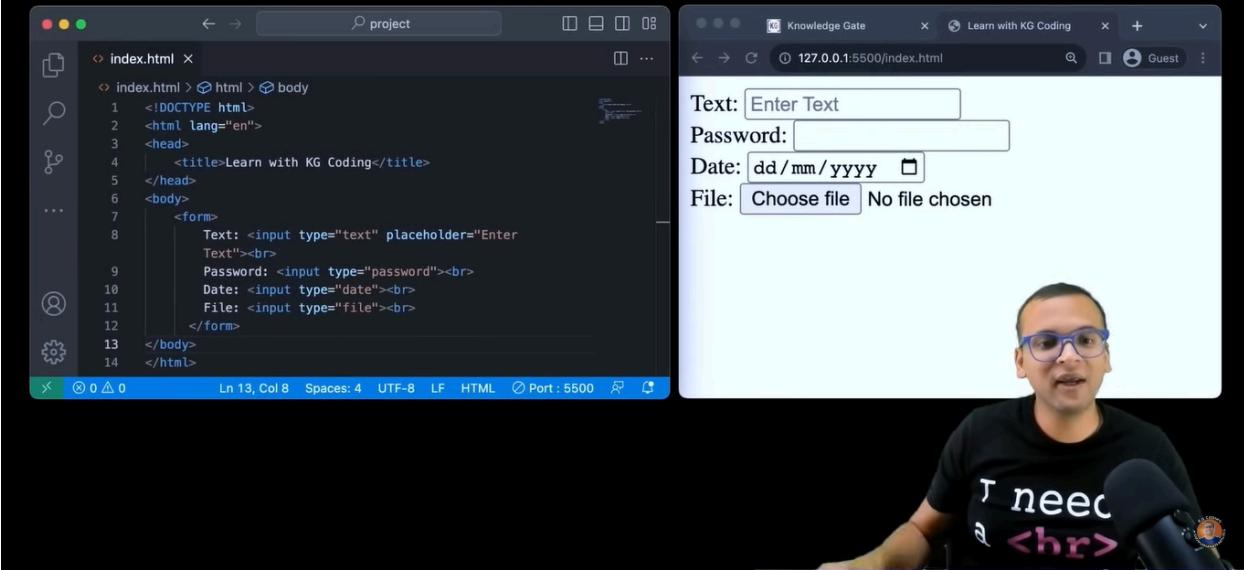
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Tag</title>
</head>
<body>
    <form action="/login-data">
        <label for="user-id">Username: </label>
        <input type="text" id="user-id" placeholder="Enter Username"
required="True"> <br><br>

        <label for="password">Password: </label>
        <input type="password" id="password" placeholder="Enter Password"
required="True"> <br><br>

        <label for="date">Date:</label>
        <input type="date"><br><br>
        <input type="submit">
    </form>
</body>
</html>
```

LEVEL 5

## 3.5 Input type: File



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Tag</title>
</head>
<body>
    <form>
        Date:<input type="date"><br><br>
        Upload File: <input type="file">
    </form>
</body>
</html>
```

## LEVEL 5

# 3.5 Input type: Radio

The image shows a code editor on the left and a browser window on the right. The code editor displays the contents of index.html, which contains HTML code for a form with three radio buttons labeled 'Male', 'Female', and 'Other'. The browser window shows the rendered form with the 'Male' radio button selected. Below the browser window, a video player interface is visible, showing a person wearing glasses and a t-shirt with text on it.

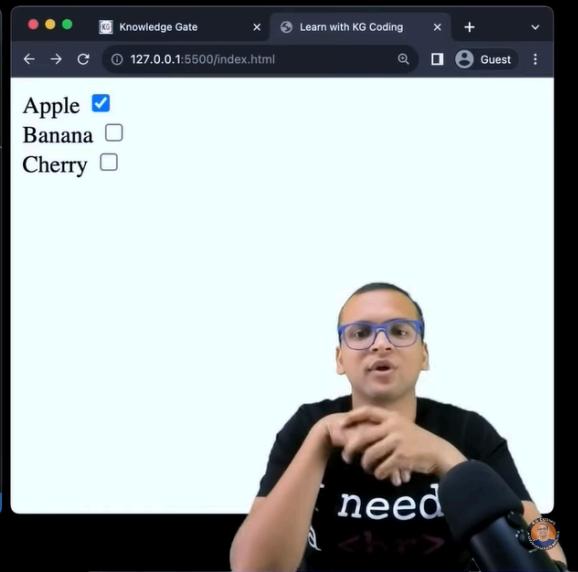
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Forms</title>
</head>
<body>
    <form>
        <label for="male"> Male:</label>
        <input type="radio" id="male" name="gender" value="Male">
        <label for="female"> Female:</label>
        <input type="radio" id="female" name="gender" value="Female">
        <label for="other"> Other:</label>
        <input type="radio" id="other" name="gender" value="Other">
    </form>
</body>
</html>
```

For from label and id is used to connect radio button with "Male:" text, so even we click on male it get select, dont need to click on button always

Type is radio and name is gender so radio button will select any value where name is gender "Value" is when we click on that button that value get stored

**LEVEL 5**

## 3.5 Input type: Checkbox



The image shows a screenshot of a video call interface. On the right, a person wearing glasses and a black t-shirt with the word "need" is speaking. They have their hands clasped under their chin. On the left, there is a code editor window showing an HTML file named "index.html". The code contains three checkbox inputs labeled "Apple", "Banana", and "Cherry". The "Apple" checkbox is checked. Below the code editor, a terminal window shows the command "Port : 5500". To the right of the terminal, a browser window titled "Knowledge Gate" displays the rendered HTML with the checked checkbox.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Checkbox</title>
</head>
<body>
    <form>
        <label for="apple">Apple</label>
        <input type="checkbox" id="apple" name="fruit" value="Apple"><br>

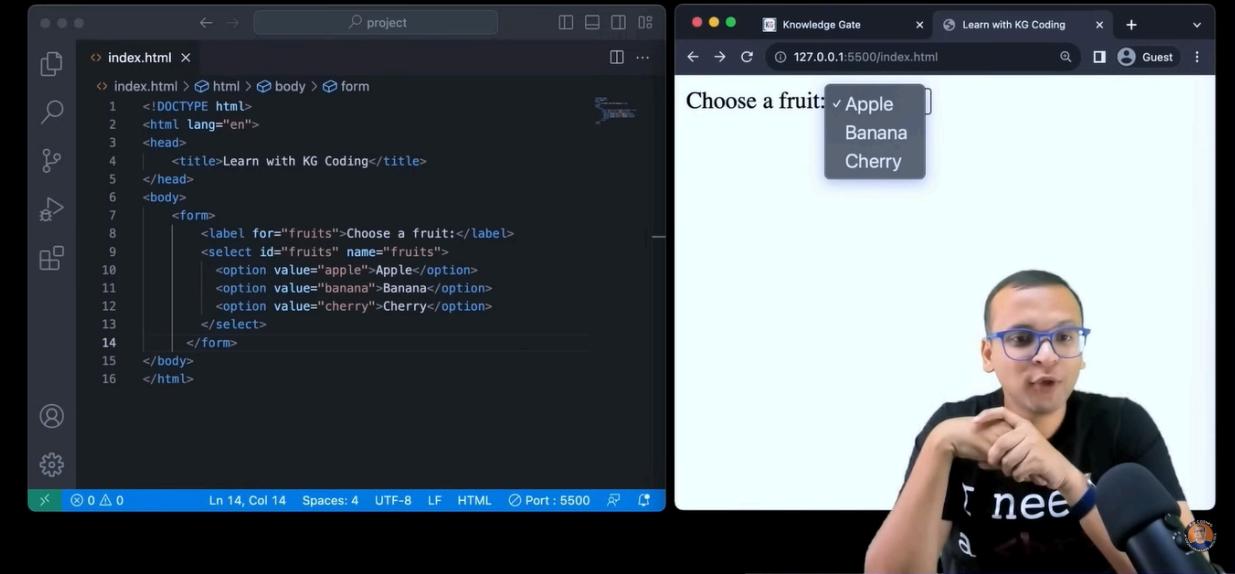
        <label for="banana">Banana</label>
        <input type="checkbox" id="banana" name="fruit" value="Banana"><br>

        <label for="cherry">Cherry</label>
        <input type="checkbox" id="cherry" name="fruit" value="Cherry"><br>
    </form>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Checkbox</title>
</head>
<body>
    <form>
        <label for="ajinkya">Ajinkya</label>
        <input type="checkbox" name="name" id="ajinkya" value="Ajinkya"><br>
        <label for="chiku">Varesh</label>
        <input type="checkbox" name="name" id="chiku" value="Varesh"><br>
        <label for="viren">Viren</label>
        <input type="checkbox" name="name" id="viren" value="Viren"><br>
        <label for="ankush">Ankush</label>
        <input type="checkbox" name="name" id="ankush" value="Ankush"><br>
        <label for="bapu">Amit</label>
        <input type="checkbox" name="name" id="bapu" value="Amit">
    </form>
</body>
</html>
```

## LEVEL 5

# 3.5 Input type: Select



The image shows a developer's environment with two windows. On the left is a code editor displaying the file 'index.html' with the following code:

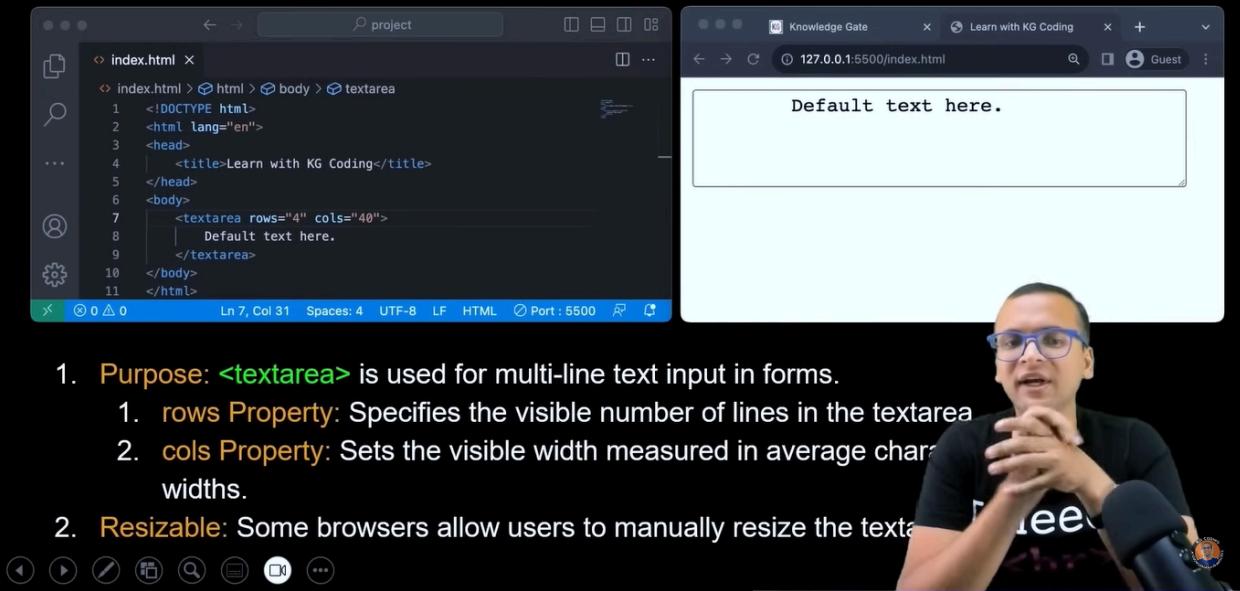
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form>
        <label for="fruits">Choose a fruit:</label>
        <select id="fruits" name="fruits">
            <option value="apple">Apple</option>
            <option value="banana">Banana</option>
            <option value="cherry">Cherry</option>
        </select>
    </form>
</body>
</html>
```

On the right is a browser window titled 'Knowledge Gate' showing the rendered HTML. A dropdown menu is open with the options 'Apple', 'Banana', and 'Cherry'. Below the browser window, there is a video feed of a person wearing glasses and a black t-shirt, gesturing with their hands.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>List</title>
</head>
<body>
    <form>
        <label for="name">Choose a Name</label>
        <select id="name" name="Name">
            <option value="ajinkyaa">Ajinkyaa</option>
            <option value="chiku">Varesh</option>
            <option value="amit">Amit</option>
            <option value="viren">Viren</option>
            <option value="ankush">Ankush</option>
        </select>
    </form>
</body>
</html>
```

**LEVEL 5**

## 3.5 Input type: TextArea



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <textarea rows="4" cols="40">
        Default text here.
    </textarea>
</body>
</html>
```

The browser window shows the rendered output: a text area with the placeholder text "Default text here.".

1. Purpose: `<textarea>` is used for multi-line text input in forms.
  1. rows Property: Specifies the visible number of lines in the textarea
  2. cols Property: Sets the visible width measured in average character widths.
2. Resizable: Some browsers allow users to manually resize the text area.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>basic form</title>
</head>
<body>
    <textarea name="eassy" id="eassy" cols="10" rows="6">
        write eassy here...
    </textarea>
</body>
</html>
```

## LEVEL 5

# 4.1 Using iFrames



The screenshot shows a video call interface. On the left is a code editor window titled "index.html" showing the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <iframe width="300" height="200" src="https://en.wikipedia.org/wiki/Main_Page"></iframe>
</body>
</html>
```

On the right is a browser window titled "Learn with KG Coding" displaying the Wikipedia main page. The URL in the address bar is "127.0.0.1:5500/index.html".

Below the video player, there is a list of three points:

1. **Embedded Content:** Allows you to embed another webpage or multimedia content within a webpage.
2. **src Attribute:** Specifies the **URL** of the content to be embedded.
3. **Dimensions:** **Width** and **height** can be set using width and height attributes.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>iFrame</title>
</head>
<body>
    <iframe width="560" height="315"
src="https://www.youtube.com/embed/o_5Nd1Y1Trw?si=QMdpWpK301L2xVD9"
title="YouTube video player" frameborder="0" allow="accelerometer,
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture;
web-share" referrerPolicy="strict-origin-when-cross-origin"
allowfullscreen></iframe>
</body>
</html>
```

## LEVEL 5

# Project Level 5

List, Tables & Forms

1. Create a **page** with all type of ordered list and one **unordered list**.
2. Create a **table** with headings, captions and a few rows. One of 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 this video to your page.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Project 5</title>
</head>
<body>
    <form action="/candidate-data">
        <ol type="a">
            <li>Read all Instruction</li>
            <li>Complete given tasks</li>
        </ol>
        <ul>
            <li>Read all Instruction</li>
            <li>Complete given tasks</li>
        </ul><hr>
        <table border="2">
            <caption>Random table</caption>
            <tr><th>Sr</th><th>Name</th><th>Surname</th><th>Age</th></tr>
            <tr><td>1</td><td>Ajinkya</td><td>Patil</td><td>26</td></tr>
            <tr><td>2</td><td>Varesh</td><td>Patil</td><td>16</td></tr>
```

```
</table><hr>
<h2>Contact</h2>
Name: <textarea name="Name" id="name" cols="30" rows="1"
placeholder="Enter your Name" required ></textarea><br>
Email:<textarea name="Email" id="email" cols="30" rows="1"
placeholder="Enter your Email" required ></textarea><br>
<label for="countrycode">Contact No:</label>
<select name="countrycode" id="countrycode">
<option value="india">+91</option>
<option value="pak">+92</option>
</select>
<textarea name="mobno" id="mobno" cols="30" rows="1"
placeholder="Enter your Mobile No" required ></textarea><br>
<form>
    Birth Date: <input type="date"><br>
    Upload CV: <input type="file" required><br>
    <input type="submit">
</form>
</form><hr>
<iframe width="560" height="315"
src="https://www.youtube.com/embed/z0n1aQ3IxWI?si=Ain6NEtH2ks6mx11"
title="YouTube video player" frameborder="0" allow="accelerometer,
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture;
web-share" referrerpolicy="strict-origin-when-cross-origin"
allowfullscreen></iframe>

</body>
</html>
```