Course Levels

Level 0 Setup & Fundamentals

Level 1 HTML Basics

Level 2 Must-Use HTML Tags

Level 3 Browser Tools

Level 4 HTML and Project Structure

Level 5 List, Tables & Forms

Level Bonus Github Pages & CodeSpace

Level 0

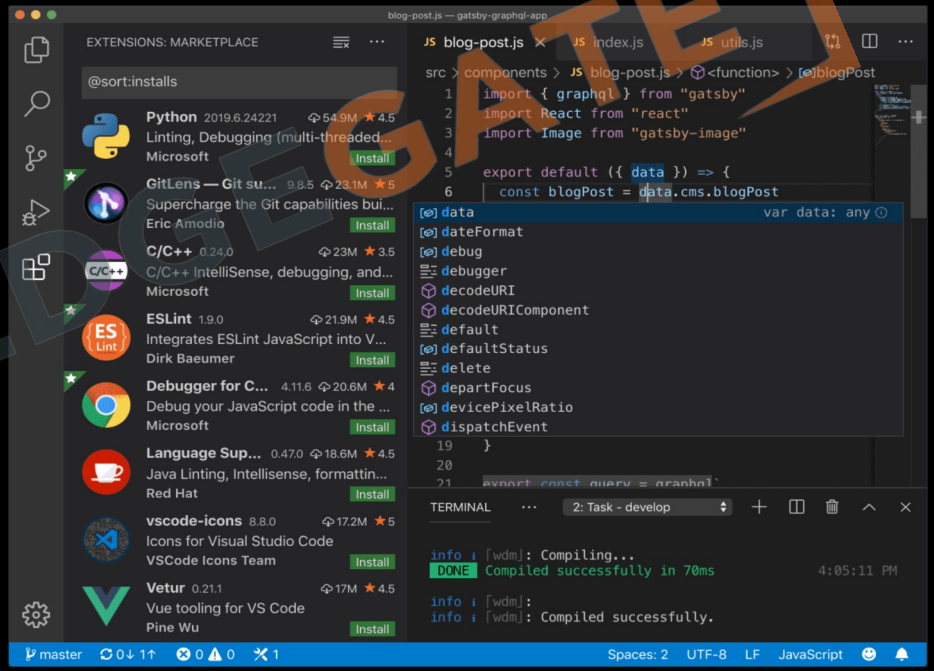
Setup & Fundamentals

1. IDE or Code Editor
    1. What is IDE
    2. Need of IDE
    3. IDE Selection
    4. Installation and Setup
    5. VsCode Extensions
2. Website Components and Fundamentals
    1. Client Side vs Server Side
    2. FrontEnd / BackEnd / FullStack
    3. Role of Browser
    4. HTML
    5. CSS
    6. JS

Level 0

Setup & Fundamentals

1. IDE
OR
Code Editor



1.1 What is IDE

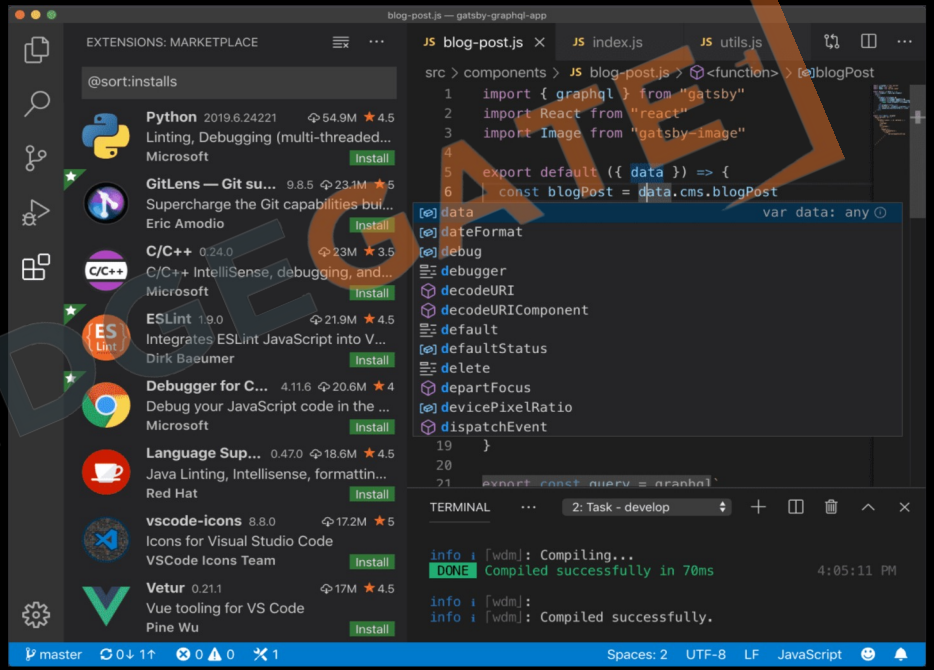
1. IDE stands for Integrated Development Environment.
2. Software suite that consolidates basic tools required for software development.
3. Central hub for coding, finding problems, and testing.
4. Designed to improve developer efficiency.

1.2 Need of IDE

1. Streamlines development.
2. Increases productivity.
3. Simplifies complex tasks.
4. Offers a unified workspace.
5. IDE Features
    1. Code Autocomplete
    2. Syntax Highlighting
    3. Version Control
    4. Error Checking

1.3 IDE Selection

1. Sublime Text
2. Atom
3. VS Code
4. Github CodeSpaces



1.4 Installation & Setup

1. Search VS Code

1.5 VsCode Extensions

1. Live Server
2. Prettier

Level 0

Setup & Fundamentals

2. Website Components And Fundamentals

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.

2.2 FrontEnd / BackEnd / FullStack

Client Side / Front-End
Web Development

Server Side
Back-End

M E R N

Full Stack

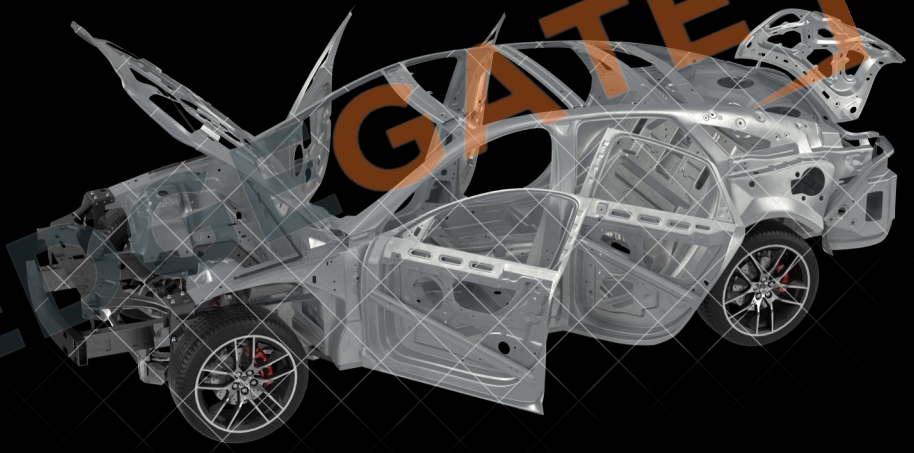
2.3 Role of Browser

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.

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.



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.

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 0 Revision

Setup & Fundamentals

1. IDE or Code Editor
    1. What is IDE
    2. Need of IDE
    3. IDE Selection
    4. Installation and Setup
    5. VsCode Extensions
2. Website Components and Fundamentals
    1. Client Side vs Server Side
    2. FrontEnd / BackEnd / FullStack
    3. Role of Browser
    4. HTML
    5. CSS
    6. JS

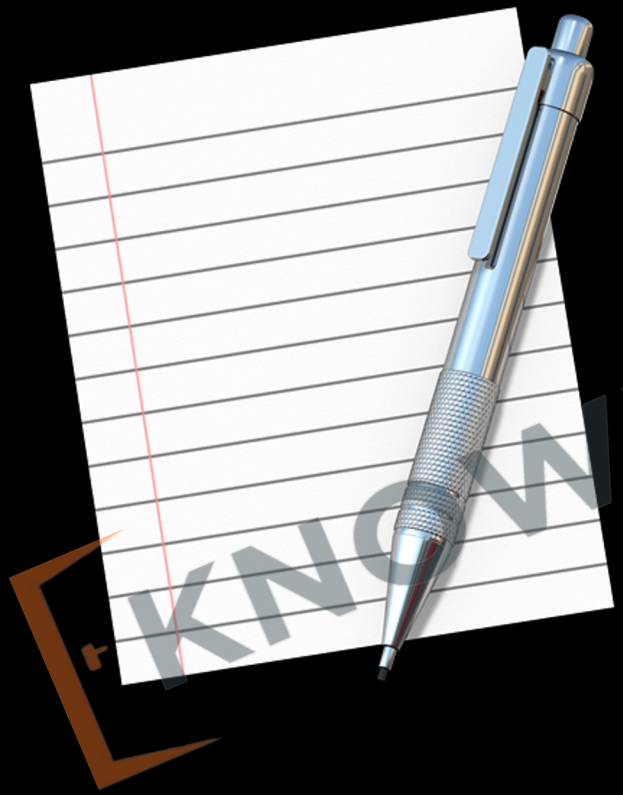
Level 1

HTML Basics

1. Starting up
    1. First File using Text Editor
    2. File Extension
    3. Opening the project in VsCode
    4. Index.html
2. Basics of HTML
    1. What are Tags
    2. Using Emmet ! to generate code
    3. Basic HTML Page
    4. MDN Documentation
    5. Comments
    6. Case Sensitivity

Level 1

HTML Basics



1. Starting Up

1.1 First file using Text Editor

1. Create a folder with name First Project on your Desktop.
2. Open Notepad.
3. Create a file and save it as index.html
4. Copy Sample code
5. Open Browser and Check.

1.2 File Extension

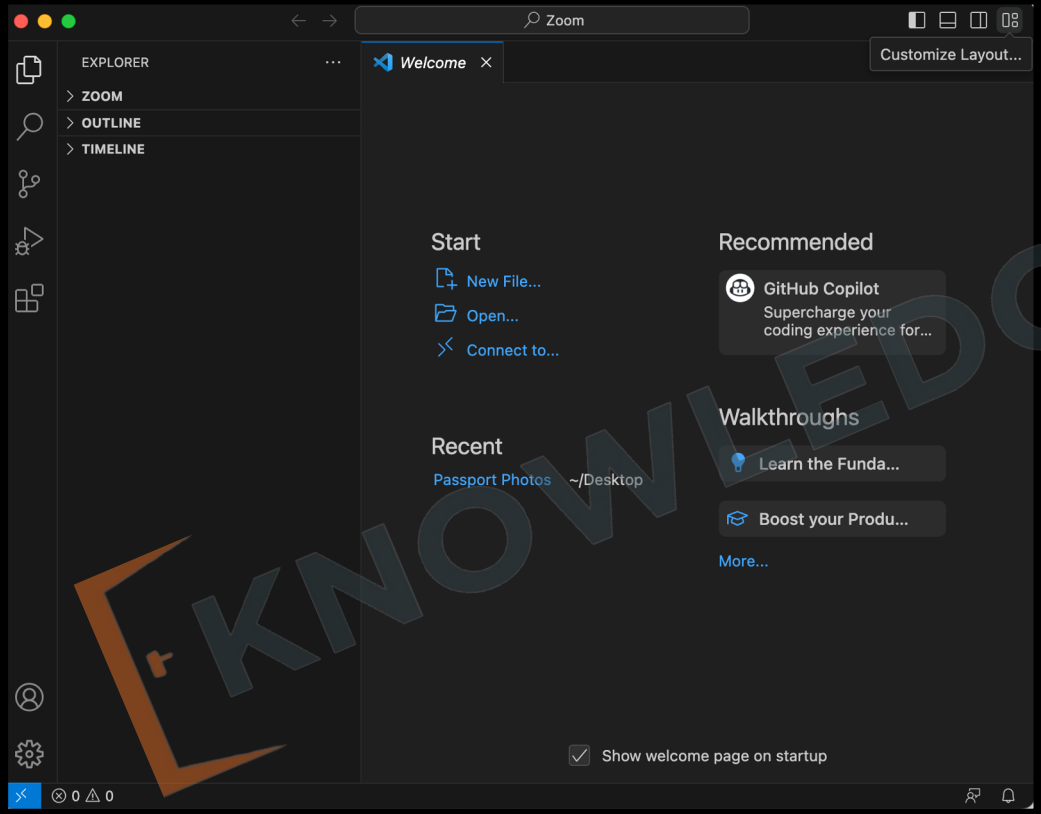
HTML

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

HTM

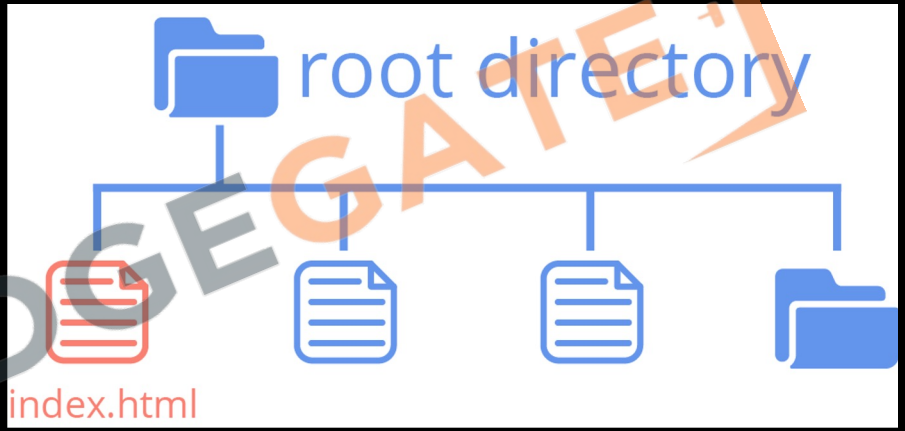
1. Less commonly used.
2. Originated for compatibility with older systems.
3. Works same as .html.
4. Typically saved as .htm.

1.3 Opening project in VsCode



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



Level 1

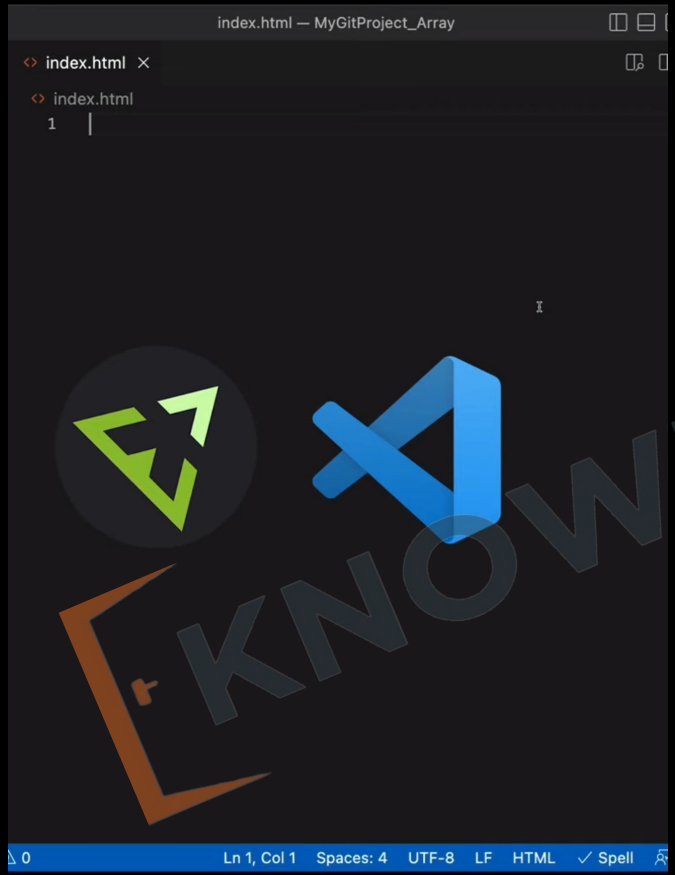
HTML Basics

2. Basics of HTML

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

2.2 Using Emmet ! to generate code



1. Type ! and wait for suggestions.

2.3 Basic HTML Page

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>My First Webpage</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
</html>
```

Defines the HTML Version

Parent of all HTML tags / Root element

Parent of meta data tags

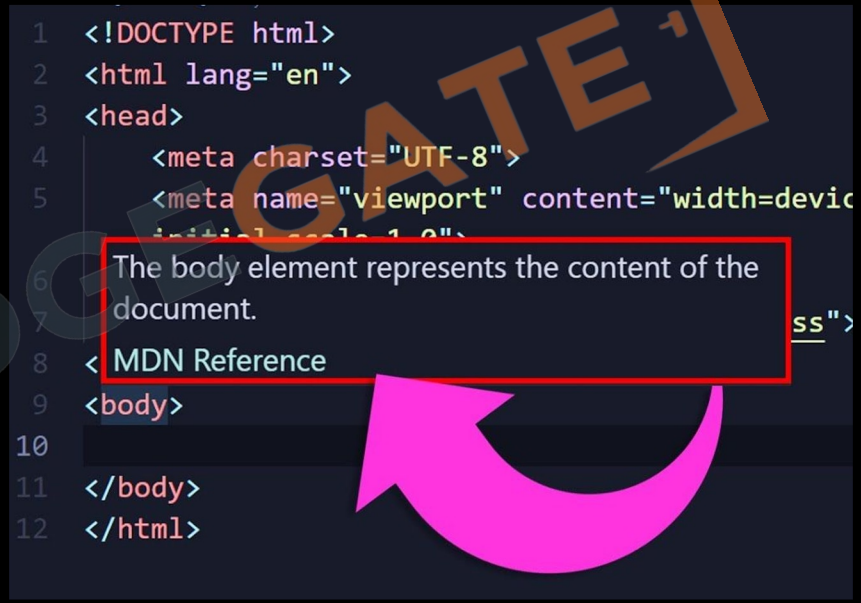
Title of the web page

Parent of content tags

Heading tag

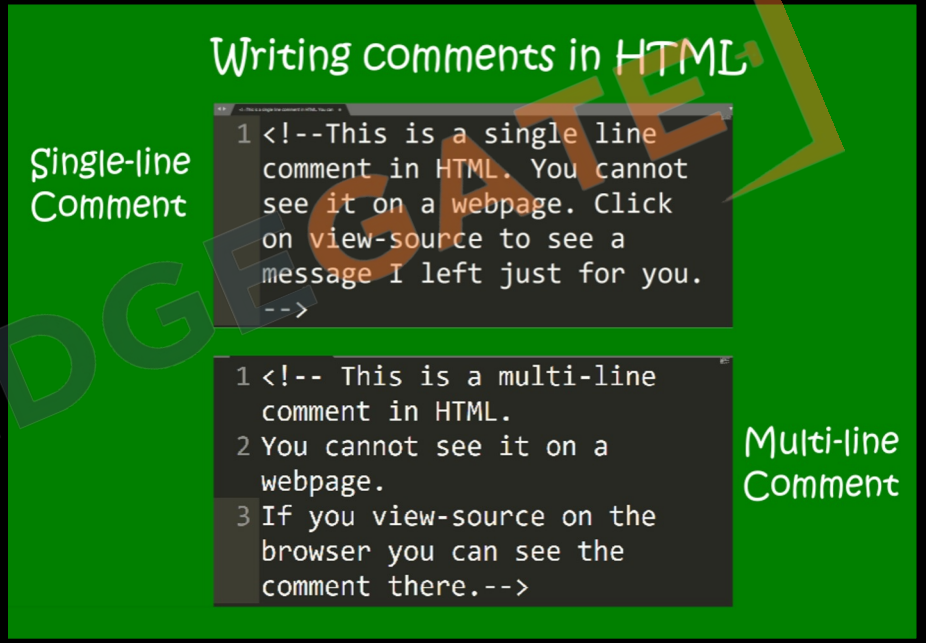
2.4 MDN Documentation

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



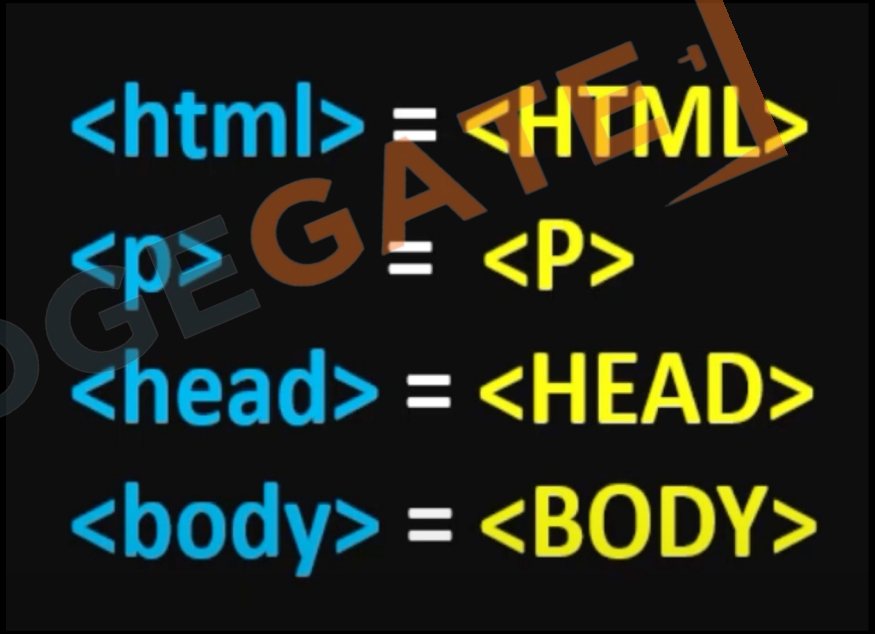
2.5 Comments

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



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



Level 1 Revision

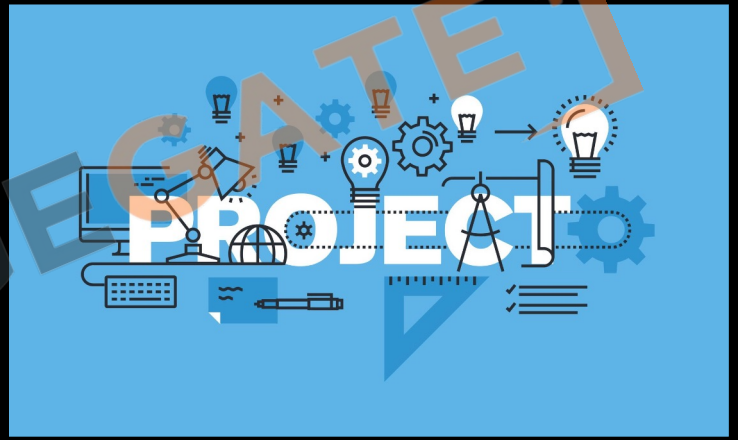
HTML Basics

1. Starting up
    1. First File using Text Editor
    2. File Extension
    3. Opening the project in VsCode
    4. Index.html
2. Basics of HTML
    1. What are Tags
    2. Using Emmet ! to generate code
    3. Basic HTML Page
    4. MDN Documentation
    5. Comments
    6. Case Sensitivity

Project Level 1

HTML Basics

1. Create a new project with Index.html
2. Generate boilerplate code using Emmet
3. Write “I am learning with Prashant sir”
4. Use comments
5. Also use Case insensitive tags



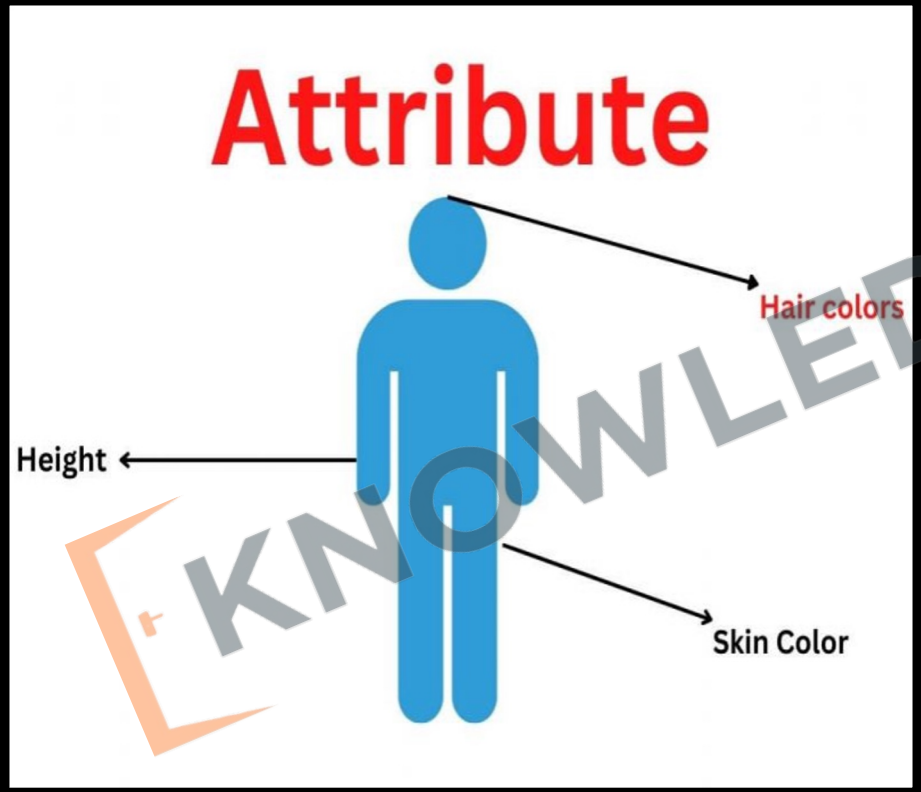
Level 2

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 References

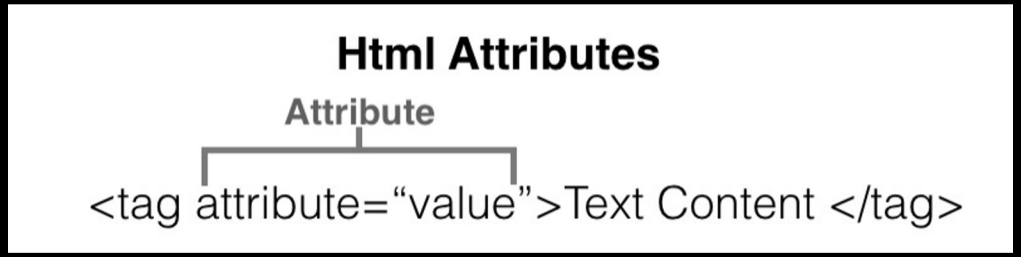
Level 2

Must-Use HTML Tags



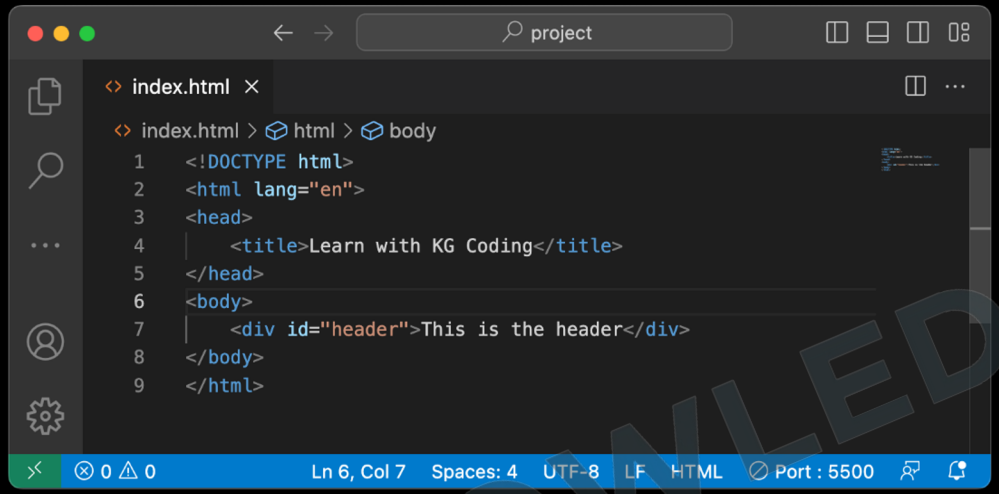
1. HTML Attributes

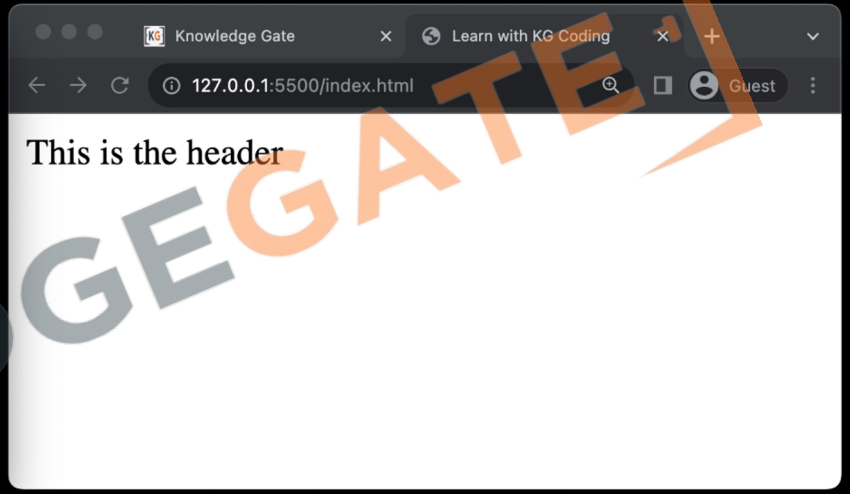
1.1 What are HTML Attributes?



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

1.2 id property





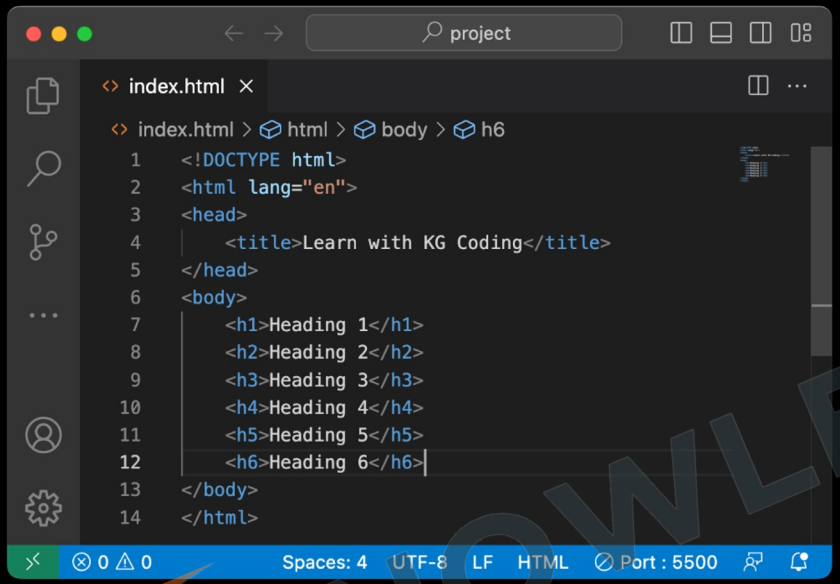
- Unique Identifier: Each id should be unique within a page.
- Anchoring: Allows for direct links to sections using the #id syntax in URLs.
- CSS & JavaScript: Used for selecting elements for styling or scripting.

Level 2

Must-Use HTML Tags

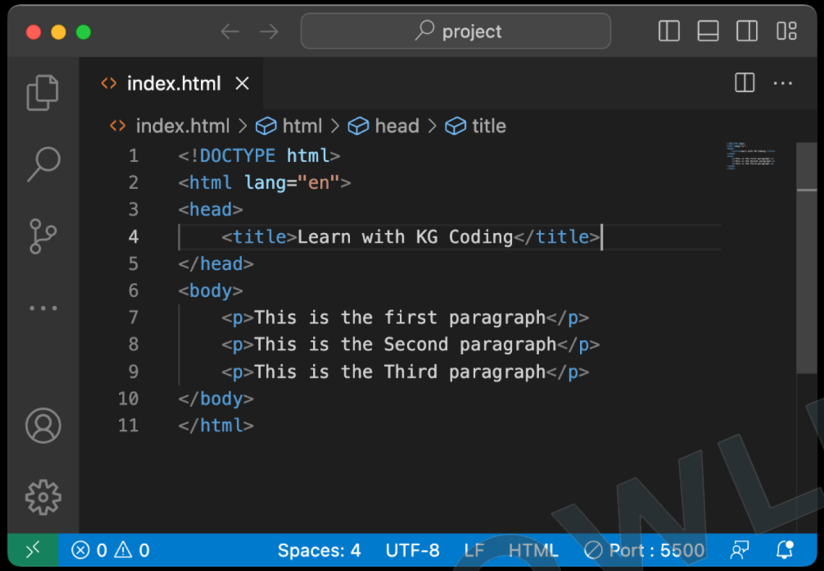
2. HTML Tags

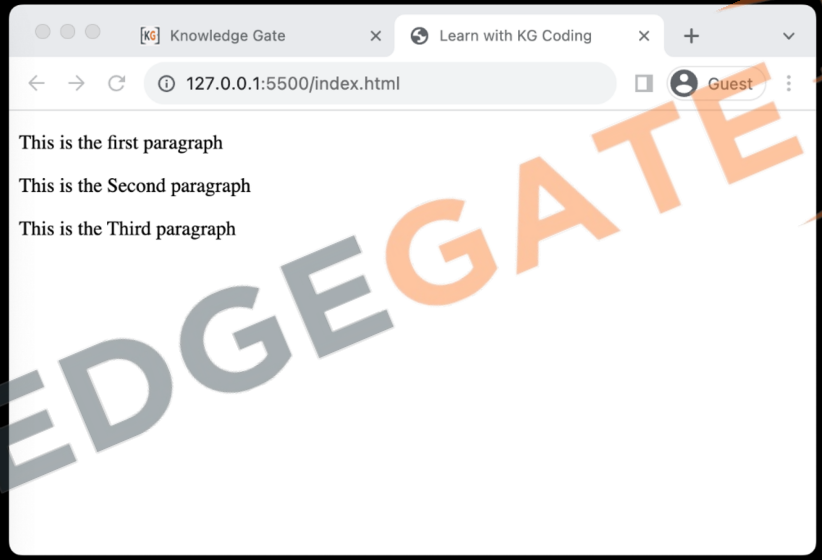
2.1 Heading Tag



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

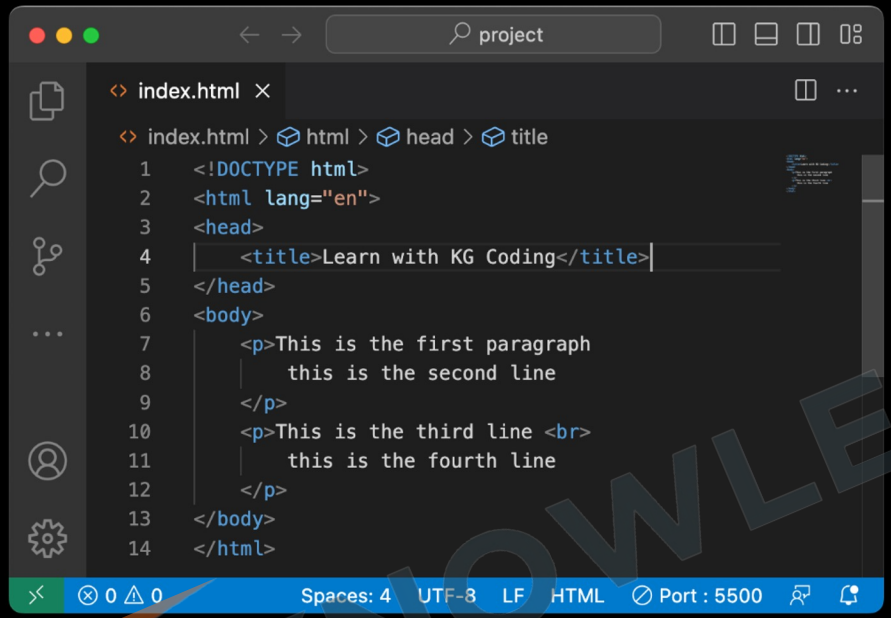
2.2 Paragraph Tag

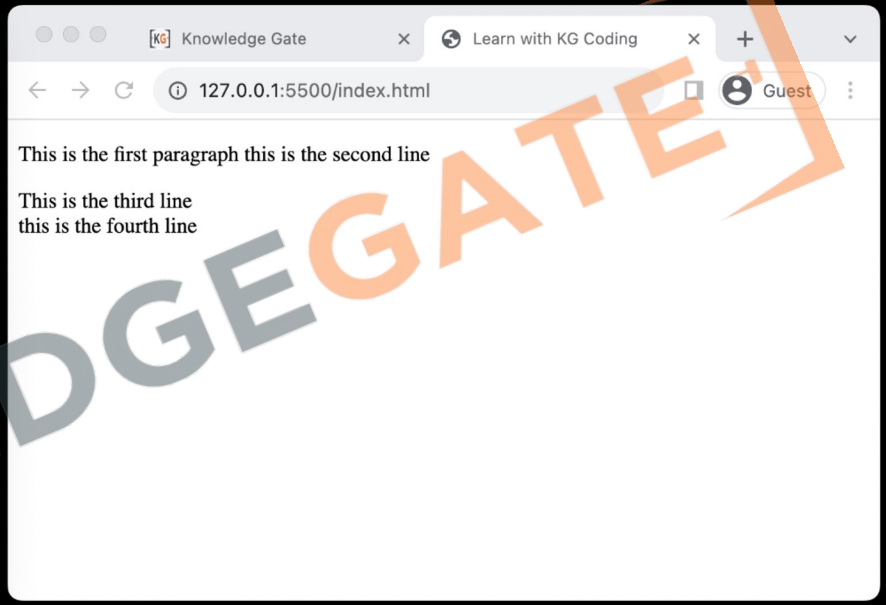




1. Used for defining paragraphs
2. Enclosed within and tags
3. Adds automatic spacing before and after
4. Text wraps to next line inside tag
5. Common in text-heavy content

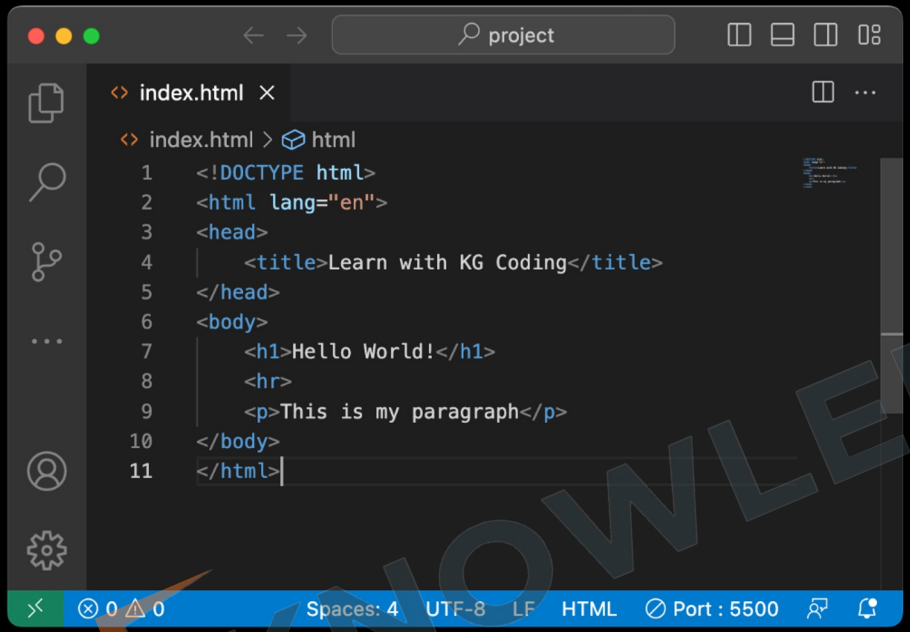
2.3
 Tag





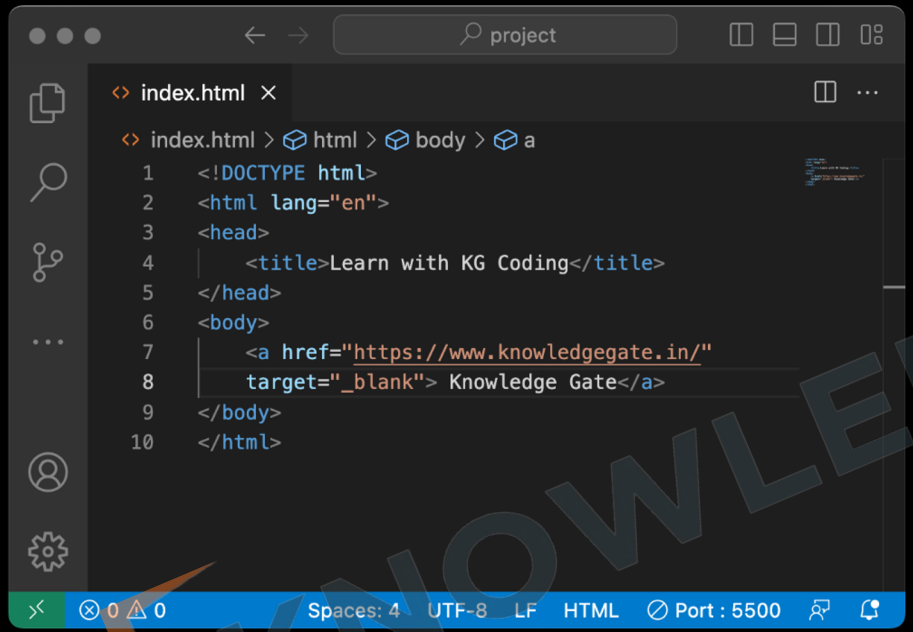
1.
 adds a line break within text
2.
 is empty, no closing tag needed
3.
 and <br /> are both valid

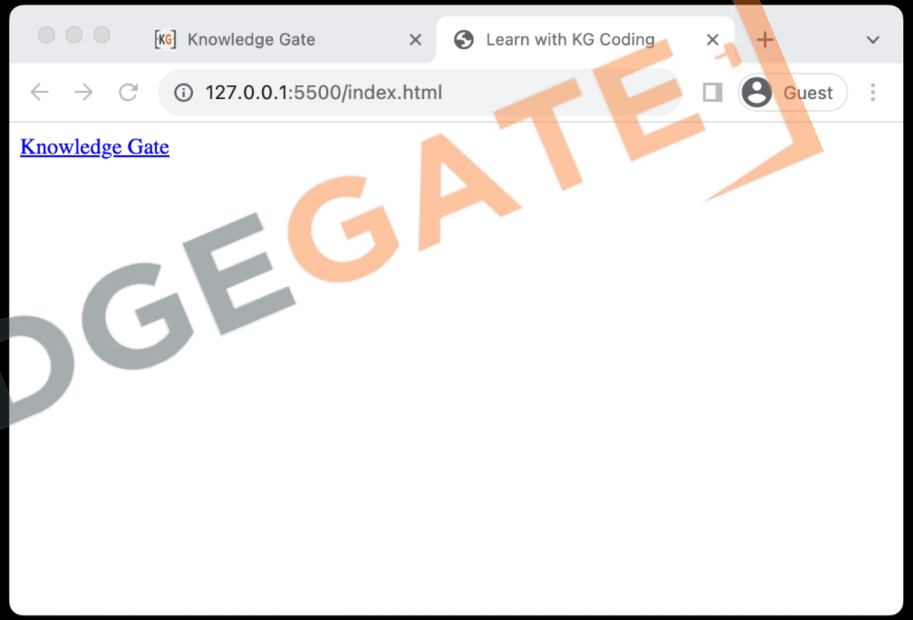
2.3 <HR> Tag



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

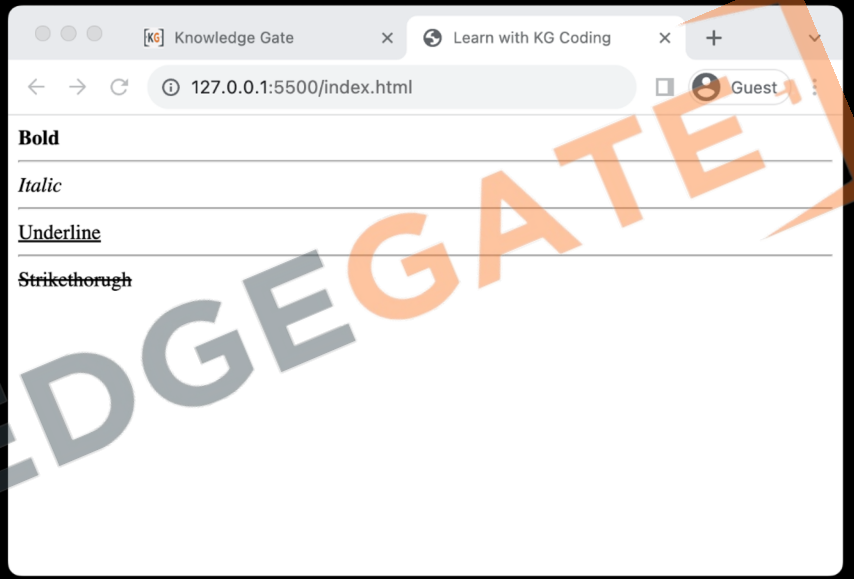
2.6 Anchor Tag





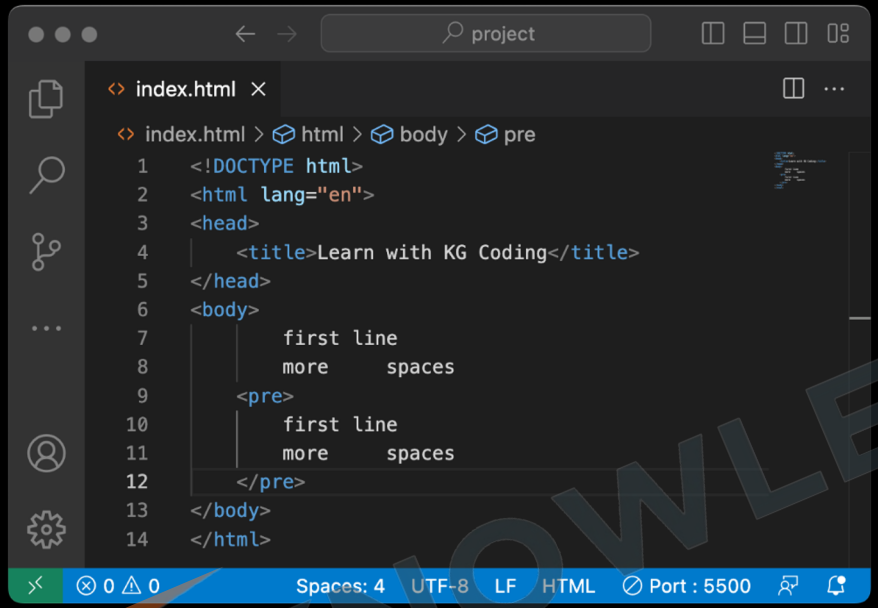
1. Used for creating hyperlinks
2. Requires href attribute for URL
3. Can link to external sites or internal pages
4. Supports target attribute to control link behavior

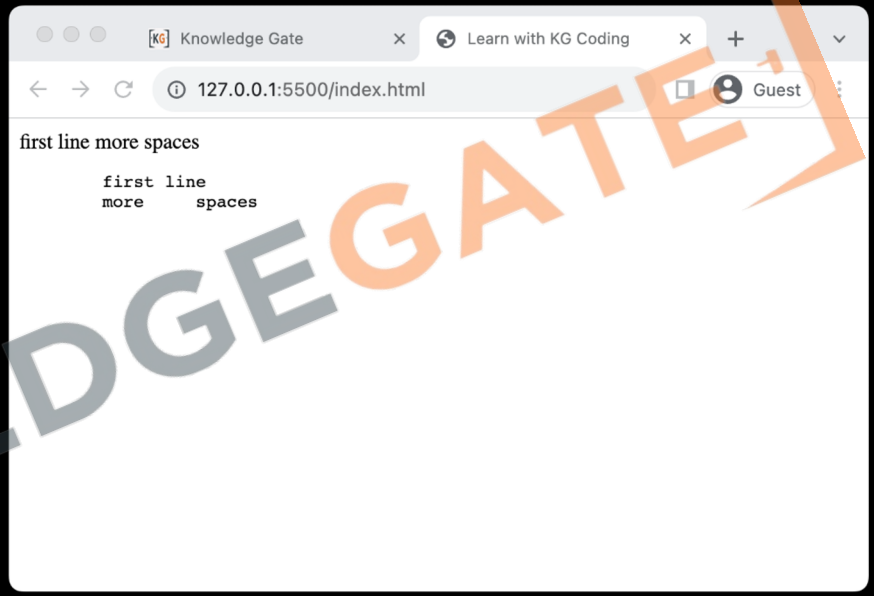
2.7 Bold/Italic/Underline/Strikethrough Tag



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

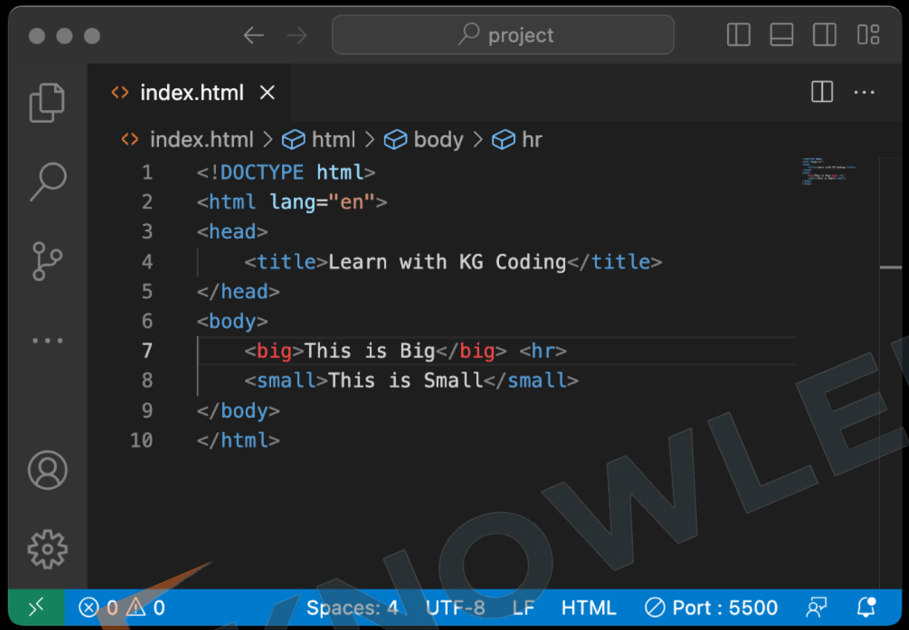
2.8 Pre Tag

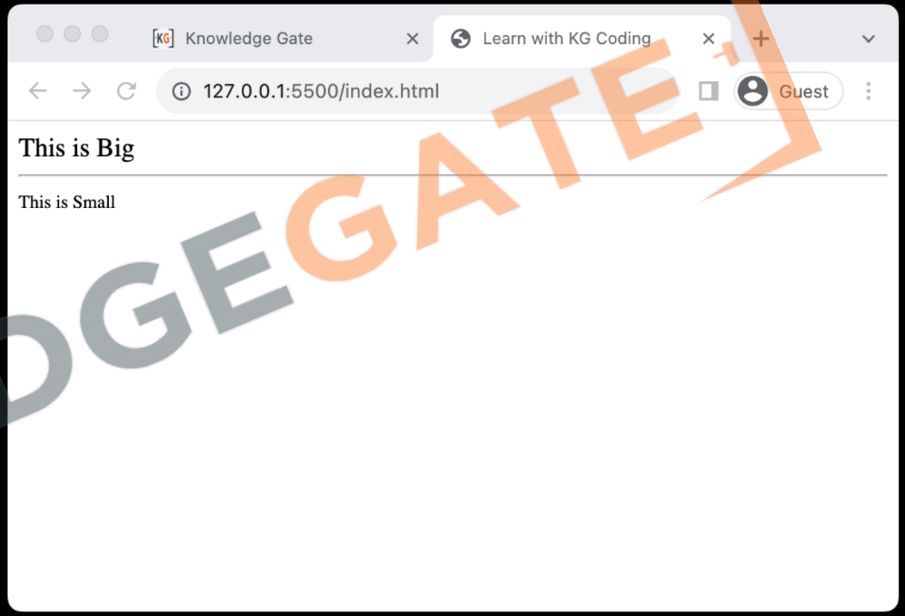




1. Preserves text formatting
2. Maintains whitespace and line breaks
3. Useful for displaying code
4. Enclosed within <pre> and </pre> tags

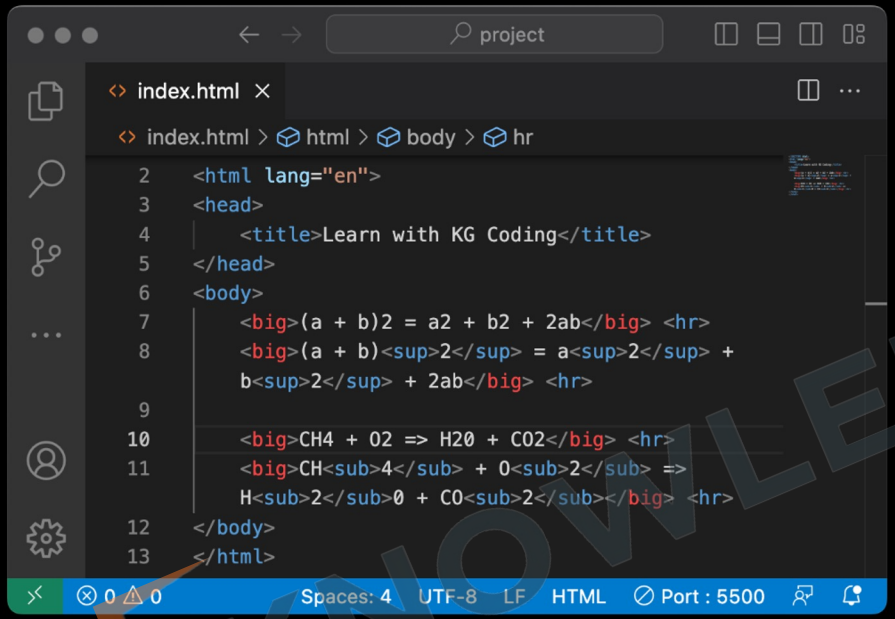
2.9 Big/Small Tag

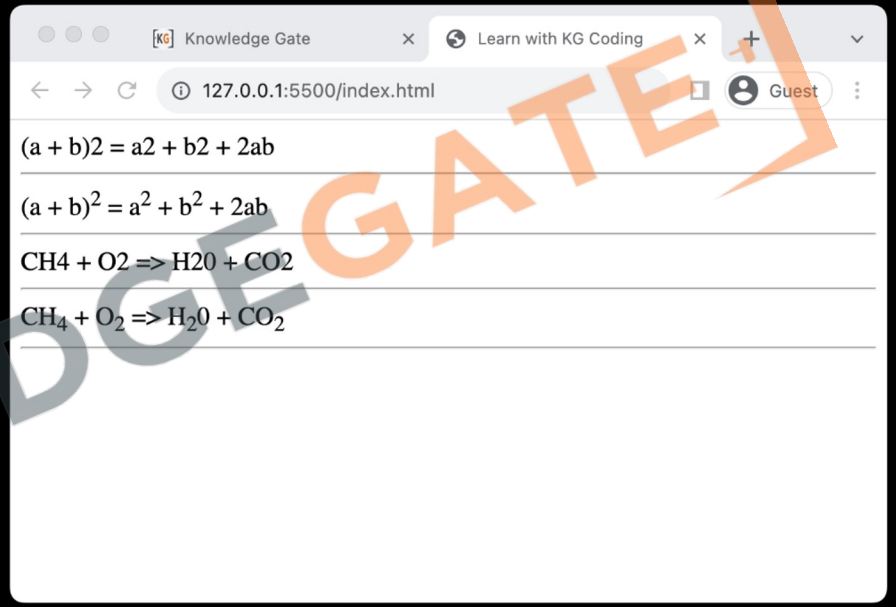




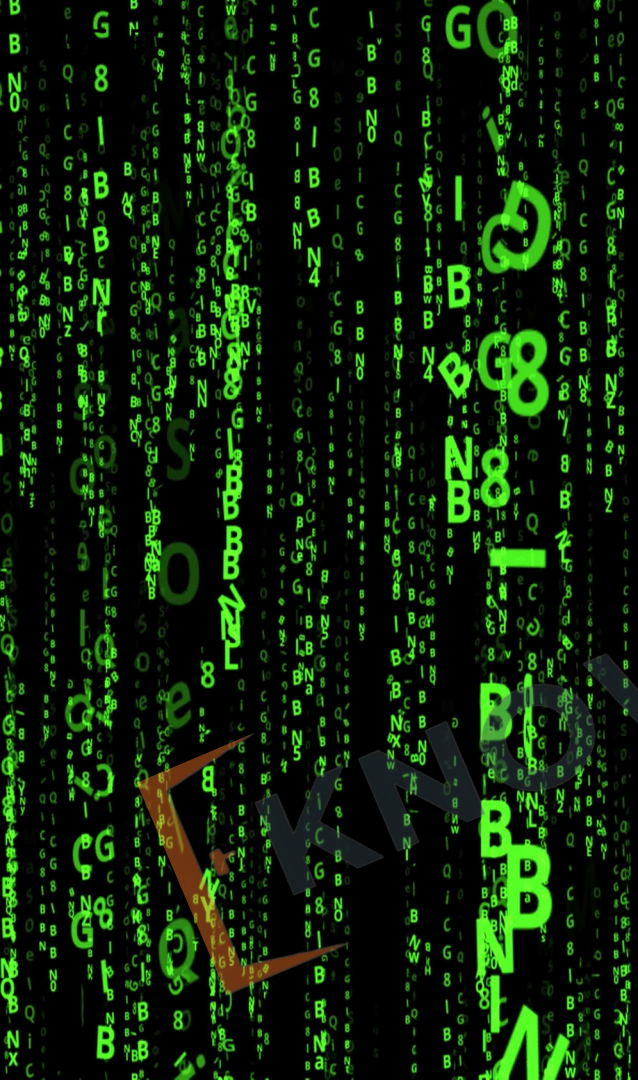
1. <big> increases text size
2. <small> decreases text size
3. Less common due to CSS alternatives

2.10 Superscript/Subscript Tag





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



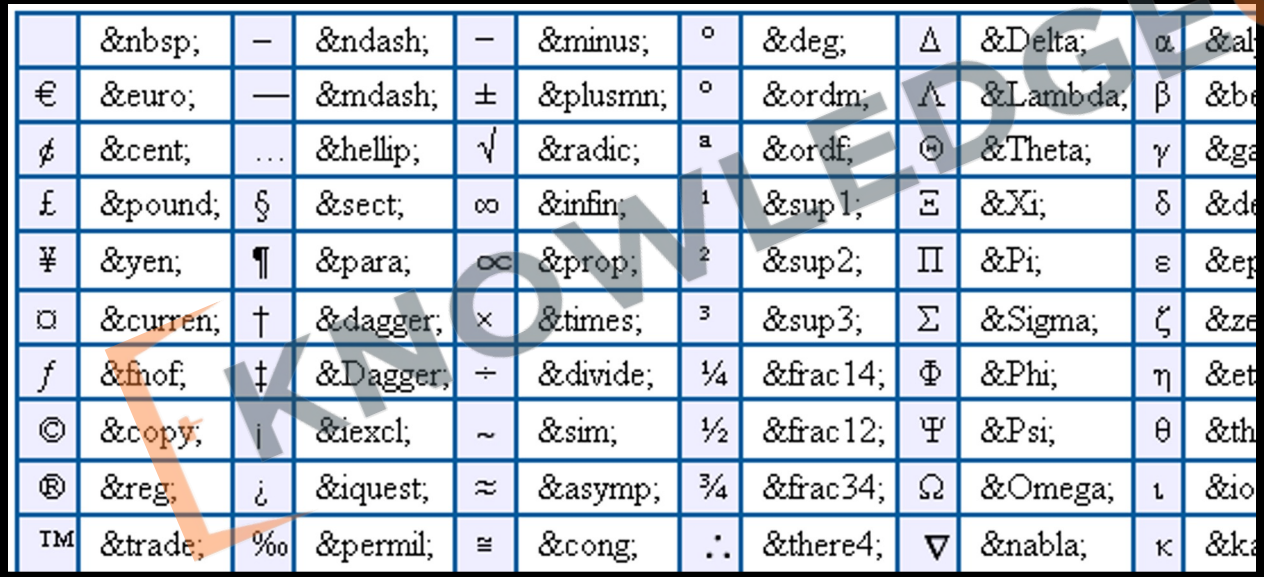
Level 2

Must-Use HTML Tags

3. Character Entity Reference

3.1 Character Entity Reference

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

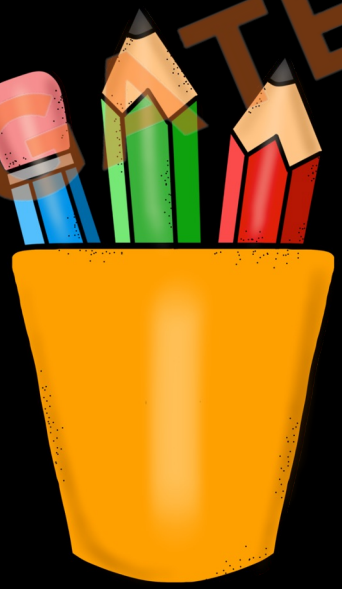


		–	&ndash;	−	&minus;	°	&deg;	Δ	&Delta;	α	&al
€	&euro;	—	&mdash;	±	&plusmn;	º	&ordm;	Λ	&Lambda;	β	&be
¢	&cent;	…	&hellip;	√	&radic;	ª	&ordf;	Θ	&Theta;	γ	&ga
£	&pound;	§	&sect;	∞	&infin;	¹	&sup1;	Ξ	&Xi;	δ	&de
¥	&yen;	¶	&para;	∝	&prop;	²	&sup2;	Π	&Pi;	ε	&ep
¤	&curren;	†	&dagger;	×	&times;	³	&sup3;	Σ	&Sigma;	ζ	&ze
ƒ	&fnof;	‡	&Dagger;	÷	&divide;	¼	&frac14;	Φ	&Phi;	η	&et
©	&copy;	¡	&iexcl;	~	&sim;	½	&frac12;	Ψ	&Psi;	θ	&th
®	&reg;	¿	&iquest;	≈	&asymp;	¾	&frac34;	Ω	&Omega;	ι	&io
™	&trade;	‰	&permil;	≅	&cong;	∴	&there4;	∇	&nabla;	κ	&ka

Level 2 Revision

Must-Use HTML Tags

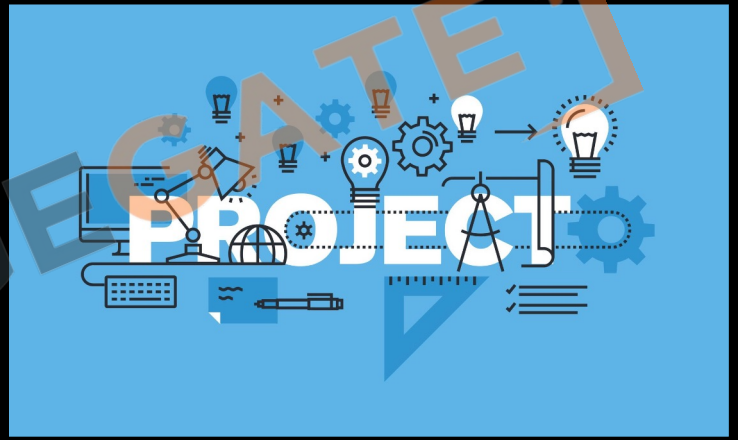
1. HTML Attributes
    1. What are HTML Attributes
    2. Id Property
2. HTML Tags
    1. Heading Tag
    2. Paragraph Tag
    3.
 <HR> tags
    4. Image Tag
    5. Video Tag
    6. Anchor Tag
    7. Bold / Italic / Underline / Strikethrough
    8. Pre Tag
    9. Big / Small Tag
    10. Superscript / Subscript
3. Character Entity Reference
    1. What are Character Entity References



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.



Level 3

Browser Tools

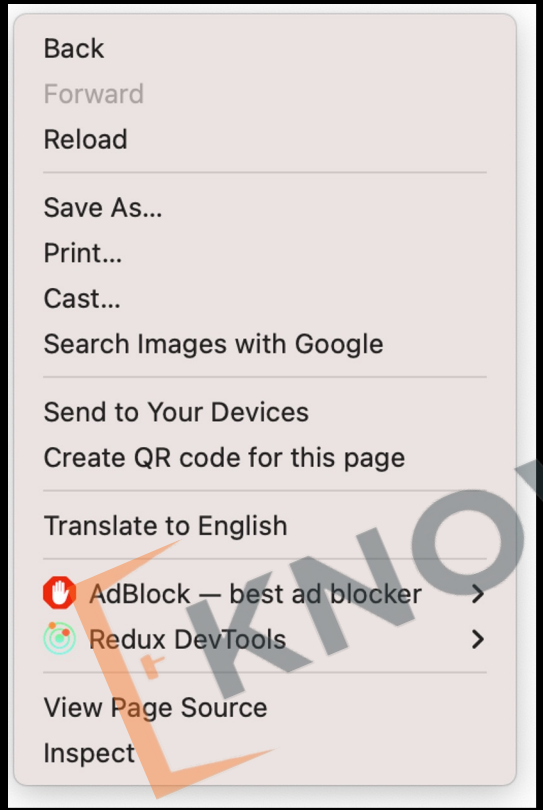
1. Browser Tools
    1. View Page Source
    2. Inspect Element
    3. HTML without CSS
2. Responsive Design
    1. Different screen size
3. Live Edit Code
    1. Live edit HTML
    2. Live edit CSS
    3. Live edit JS
    4. Changes only happening at client
4. Validating Web pages
    1. Using validator.w3.org

Level 3

Browser Tools

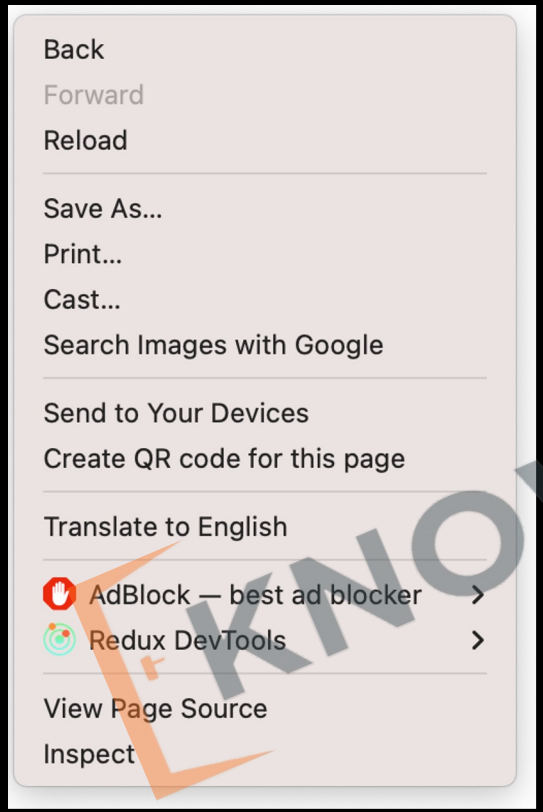
1. Browser Tools

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 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 elements on page

Level 3

Browser Tools

2. Responsive Design

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

Browser Tools

3. Live Edit Code

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

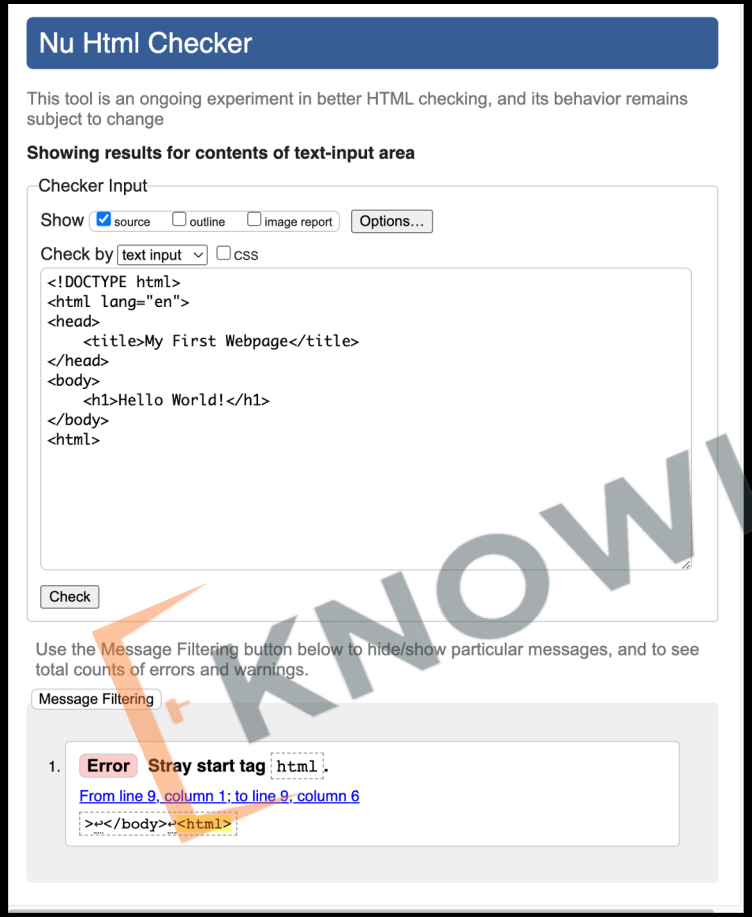
Like: If you change the question in your question paper that has no effect on actual exam.

Level 3

Browser Tools

4. Validating WebPages

4.1 Using validator.w3.org

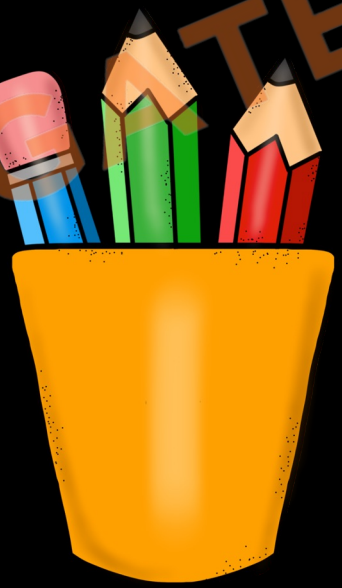


1. Ensures HTML adheres to standards
2. Minimizes cross-browser issues
3. Helps in achieving better SEO results
4. Easier to debug and maintain
5. Optimizes performance by reducing parsing errors

Level 3 Revision

Browser Tools

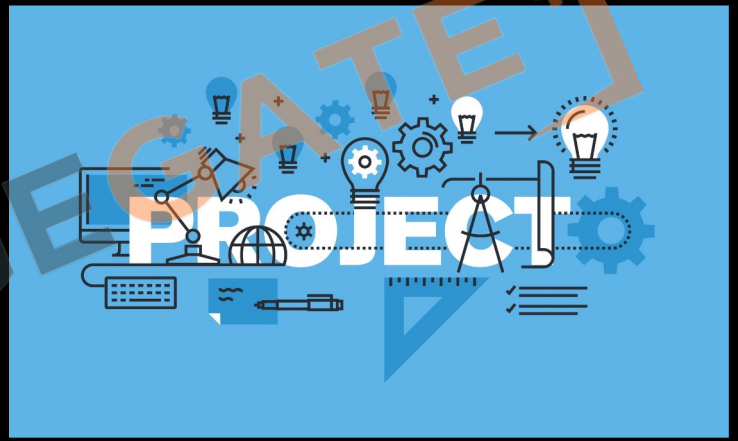
1. Browser Tools
    1. View Page Source
    2. Inspect Element
    3. HTML without CSS
2. Responsive Design
    1. Different screen size
3. Live Edit Code
    1. Live edit HTML
    2. Live edit CSS
    3. Live edit JS
    4. Changes only happening at client
4. Validating Web pages
    1. Using validator.w3.org



Project Level 3

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.



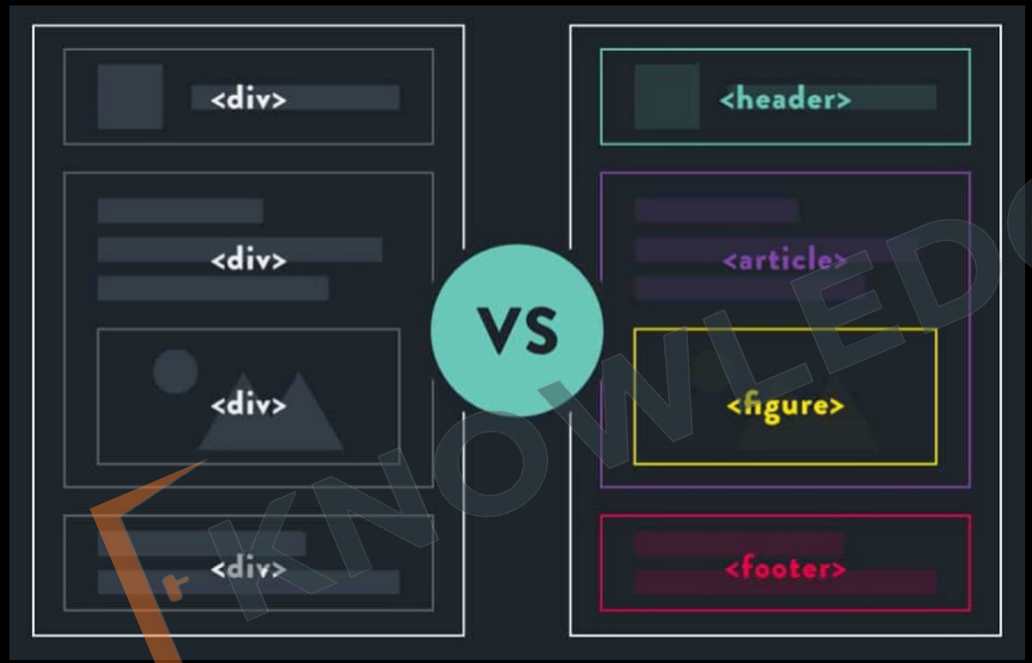
Level 4

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

HTML and Project Structure



1. Semantic Tags

1.1 Semantic/Non-Semantic Tags

Semantic Tags

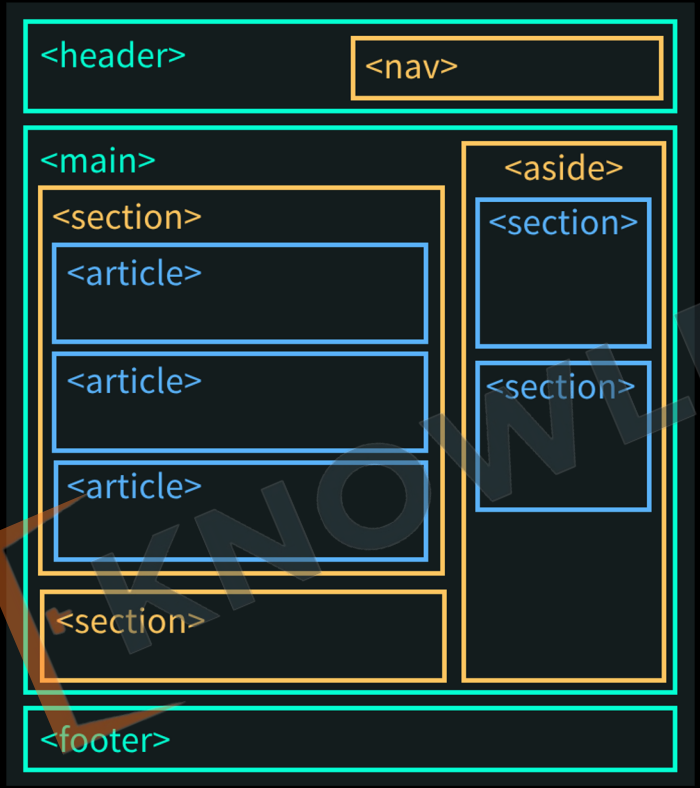
- Meaningful: Describe content.
- SEO: Good for search engines.
- Accessibility: Useful for screen readers.
- Examples: <header>, <footer>, <article>, <section>, <nav>.

Non-Semantic Tags

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

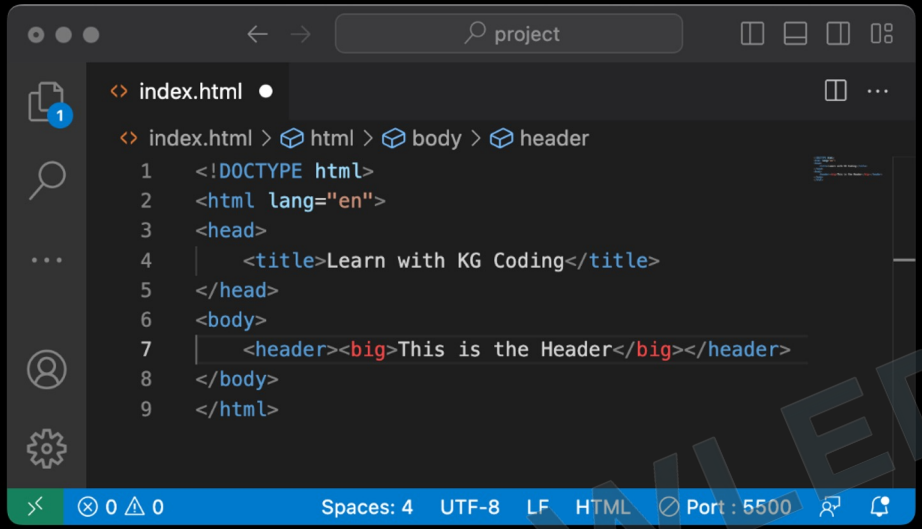
Level 4

HTML and Project Structure



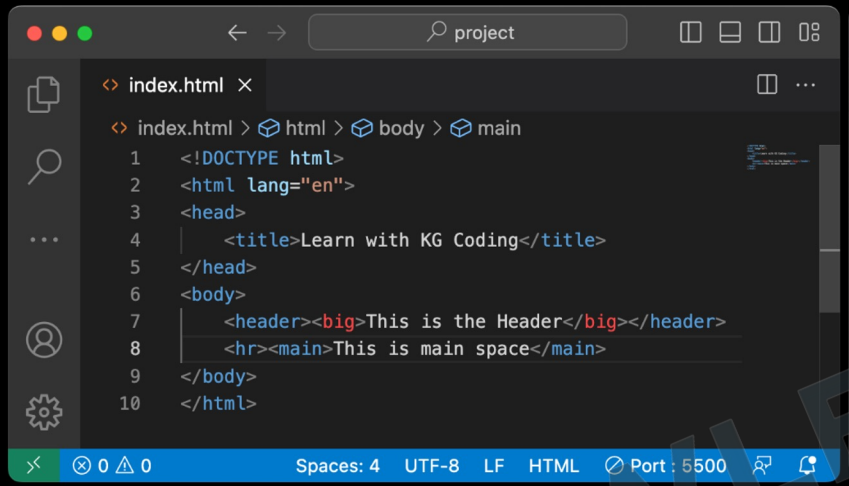
2. Body Tags

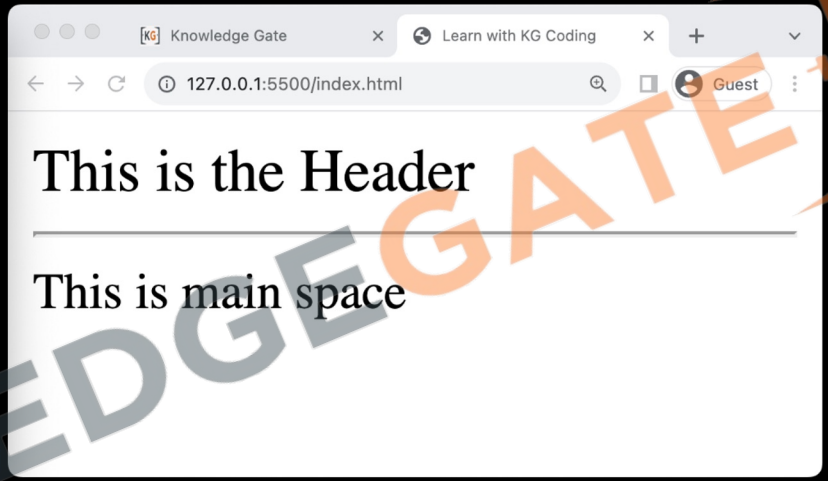
2.1 Header Tag



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.

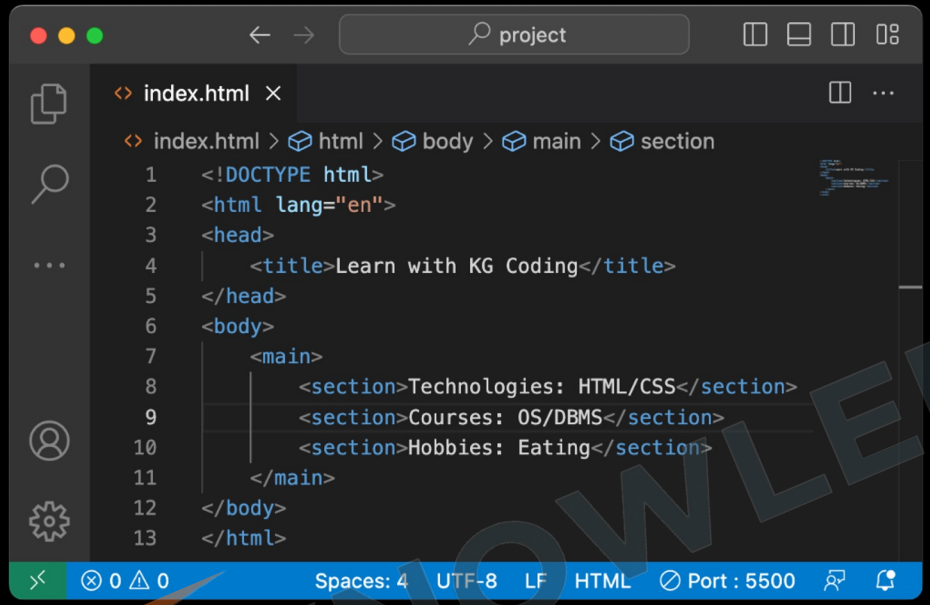
2.2 Main Tag

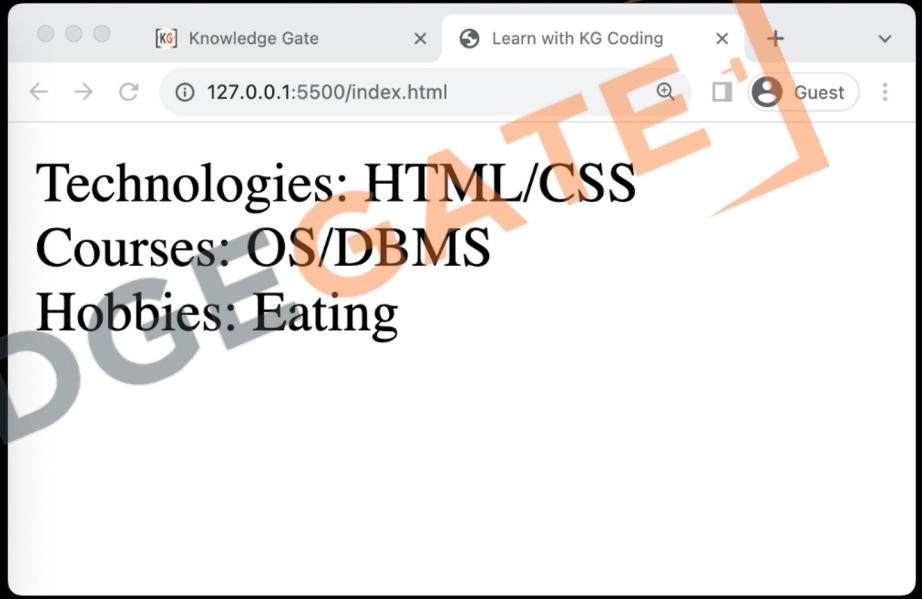




1. Purpose: Encloses the primary content of a webpage.
2. Semantic: Adds meaning, indicating the main content area.
3. Unique: Should appear only once per page.
4. Accessibility: Helps screen readers identify key content.
5. Not for Sidebars: Excludes content repeated across multiple pages like site navigation or footer.

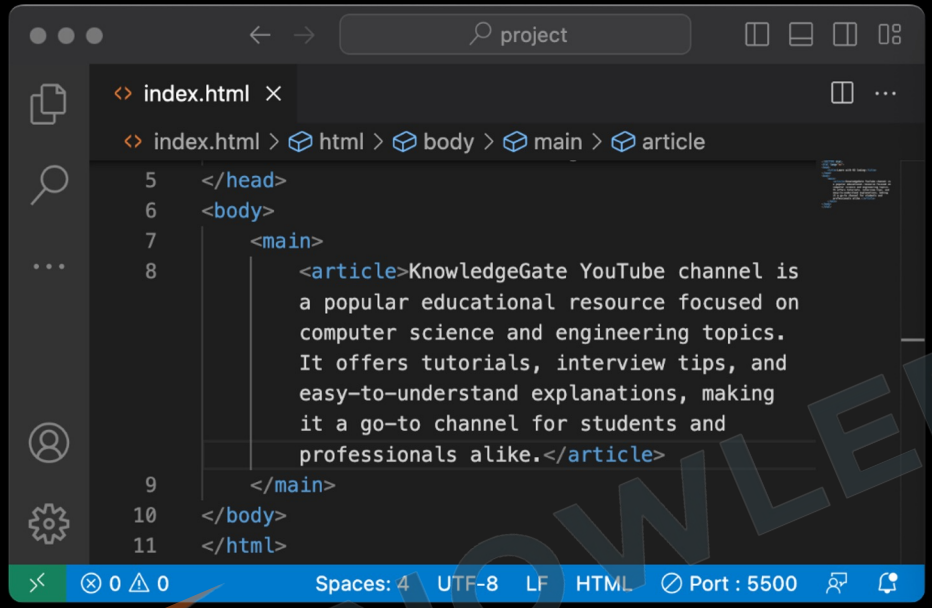
2.2.1 Section Tag

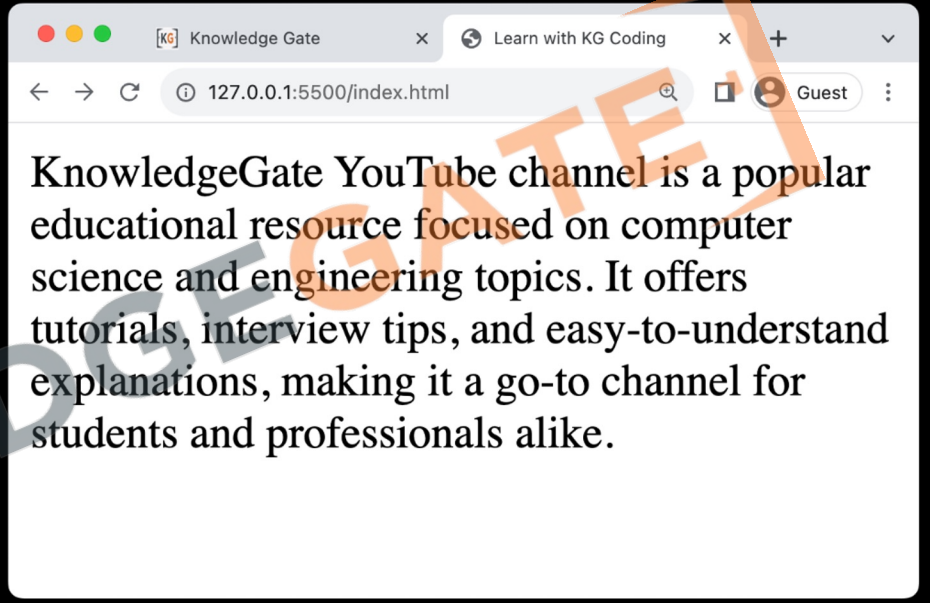




1. Purpose: Groups related content in a distinct section.
2. Semantic: Adds structure and meaning.
3. Headers: Often used with a heading <h1> to <h6> to indicate section topic.
4. Nested: Can be nested within other <section> or <article> tags.

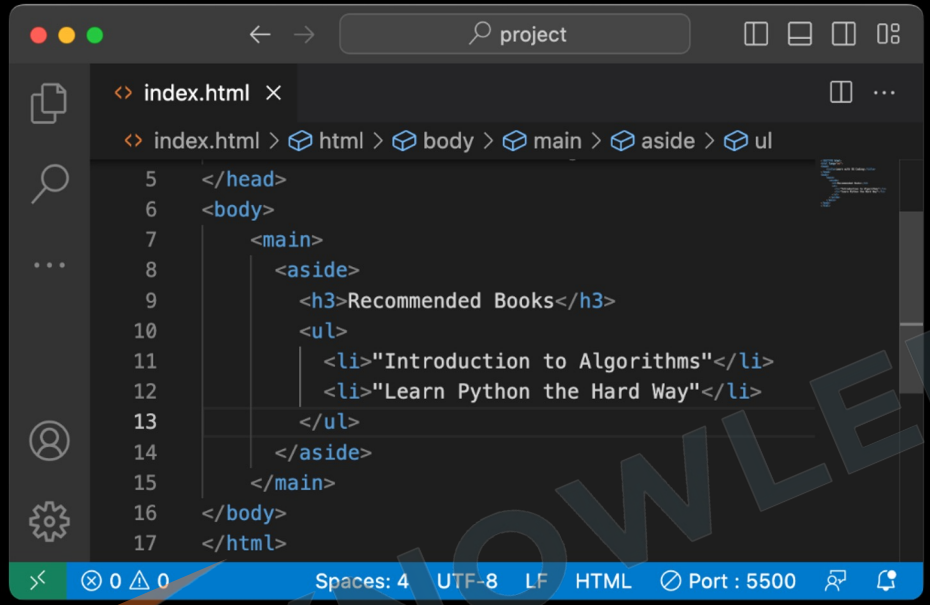
2.2.2 Article Tag

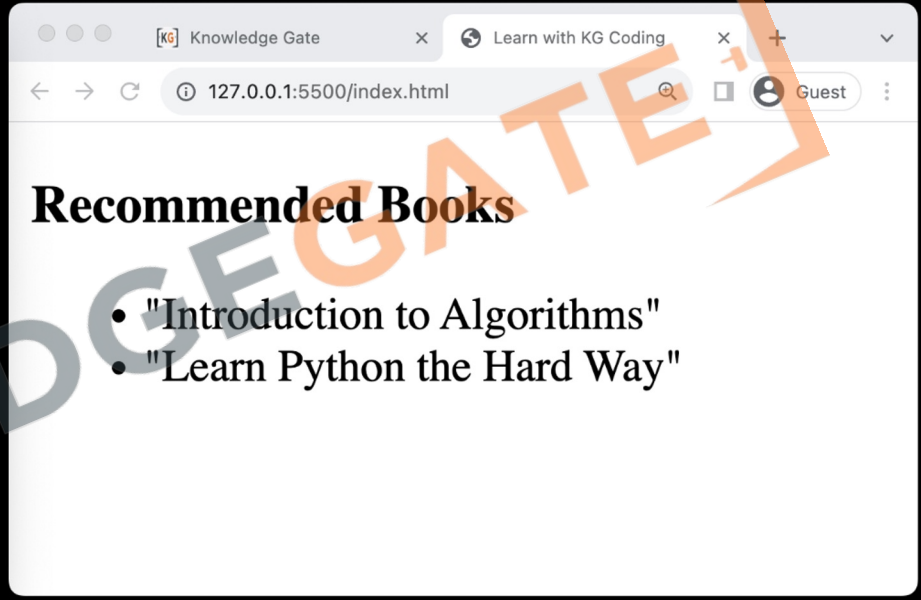




1. Purpose: Encloses content that stands alone, like a blog post or news story.
2. Semantic: Provides contextual meaning.
3. Independence: Content should make sense even if taken out of the page context.
4. Multiple Instances: Can be used multiple times on the same page

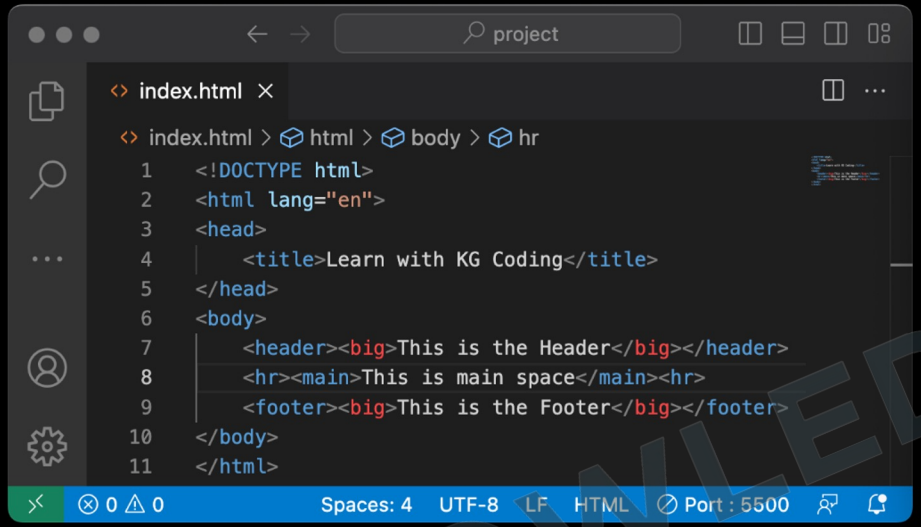
2.2.3 Aside Tag

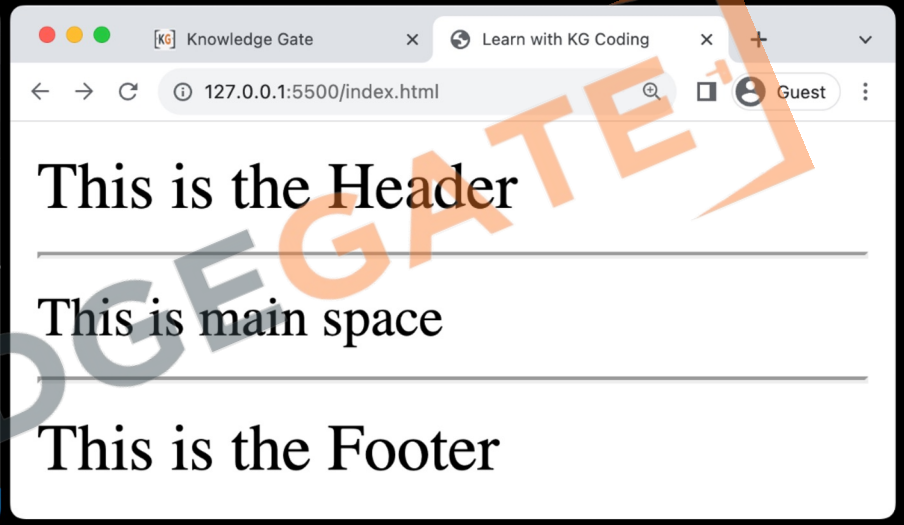




1. Purpose: Contains sidebar or supplementary content.
2. Semantic: Indicates content tangentially related to the main content.
3. Not Crucial: Content is not essential to understanding the main content.
4. Examples: Could hold widgets, quotes, or ads.

2.3 Footer Tag

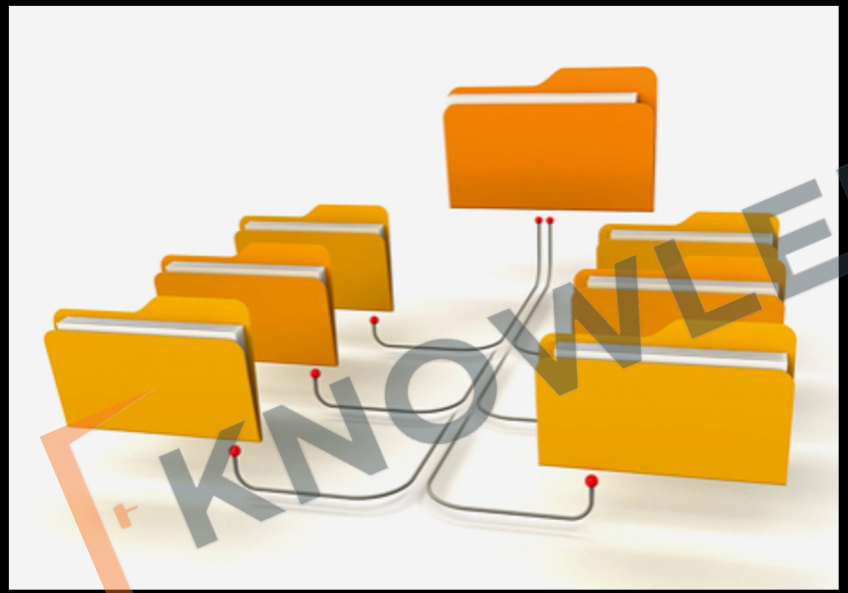




1. Purpose: For footer content like extra info or links.
2. Semantic: Provides meaning to enclosed content.
3. Location: Typically at the bottom of pages or sections.
4. Content: Includes copyrights, contact info, and social links.
5. Multiple Instances: Can be used more than once on a page.

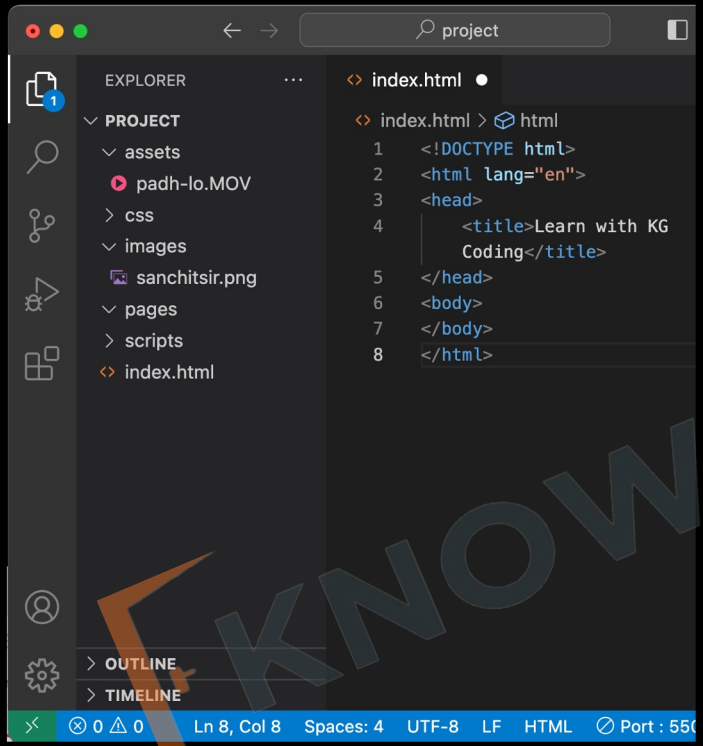
Level 4

HTML and Project Structure



3. Folder Structure

3.1 Recommended Folder Structure



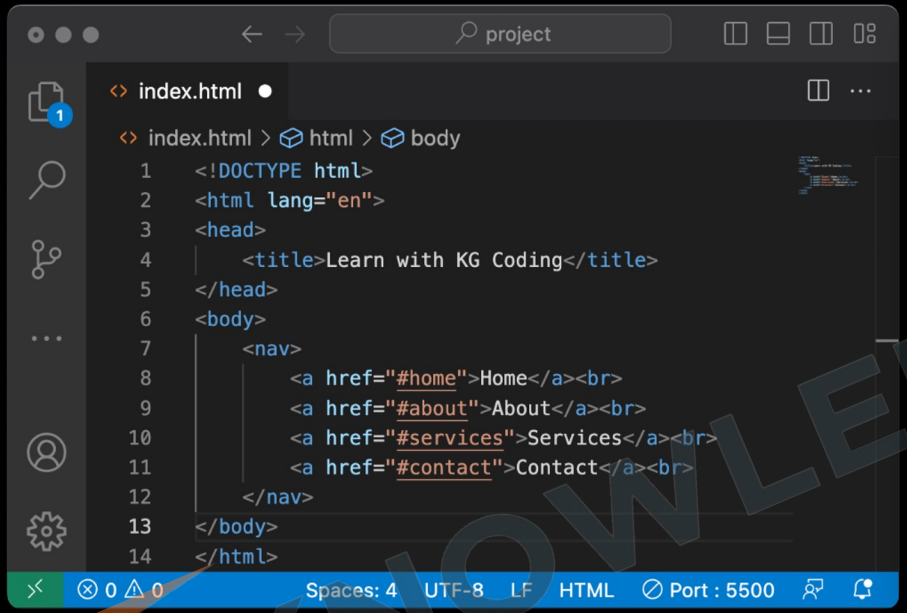
1. Root Directory: Main folder containing all website files.
2. HTML Files: Store main .html files at the root level for easy access.
3. CSS Folder: Create a css/ folder for all Cascading Style Sheets.
4. JS Folder: Use a scripts/ folder for JavaScript files.
5. Images Folder: Store images in an images/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

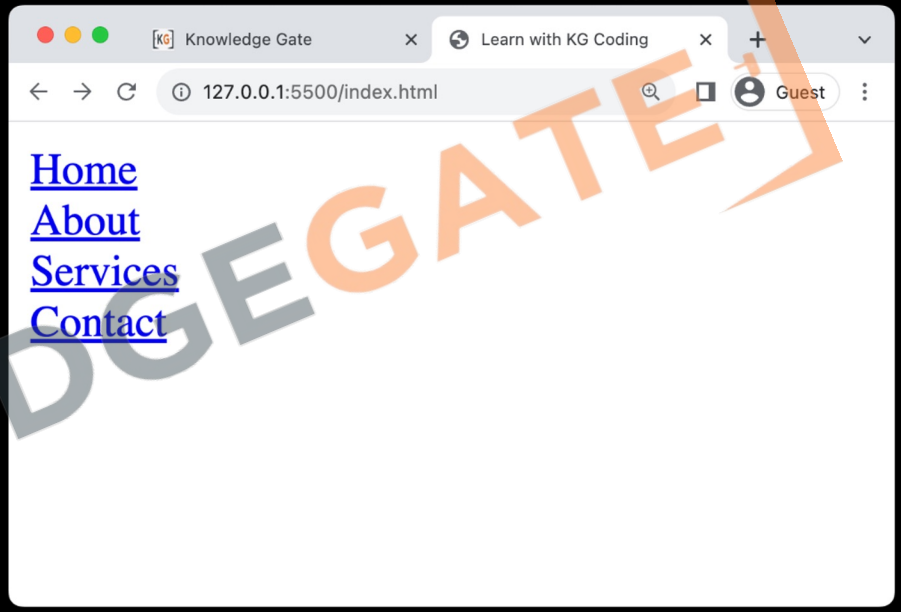
Level 4

HTML and Project Structure

4. More Tags

4.1 Navigation Tags





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.

4.2 Block / Inline Elements

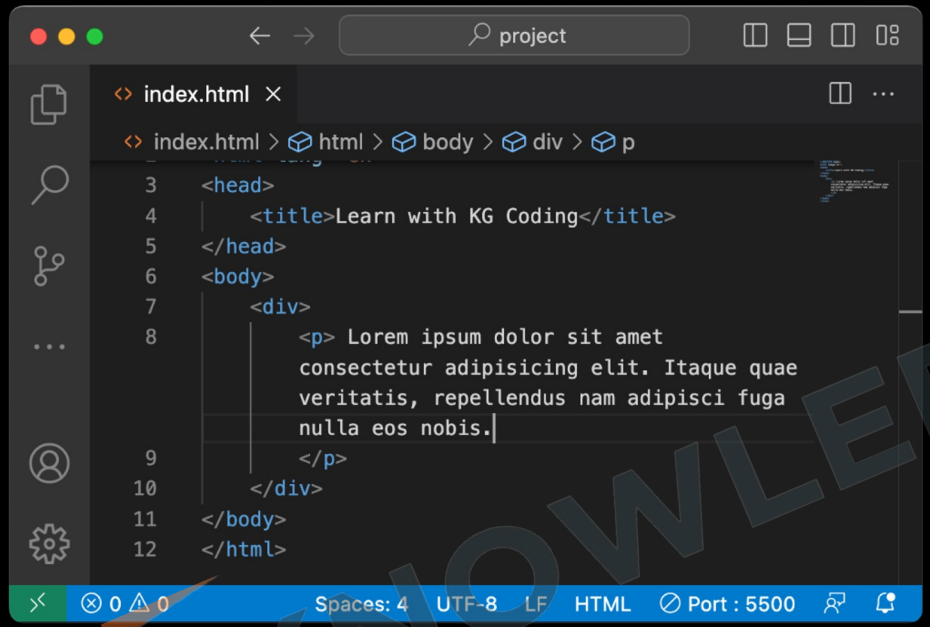
Block Elements

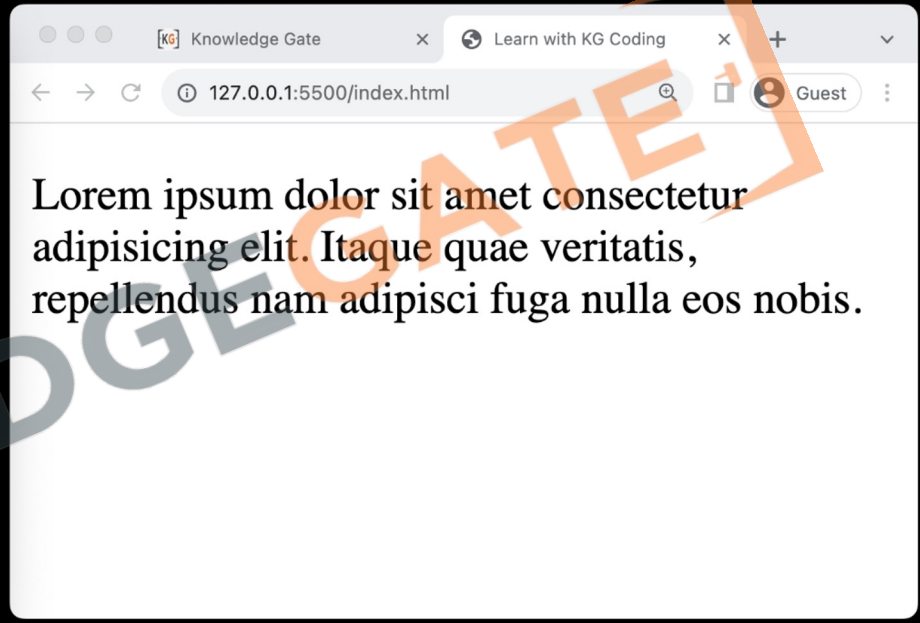
- New Line: Start on a new line.
- Full Width: Take up all horizontal space.
- Styling: Can have margins and padding.
- Size: Width and height can be set.
- Examples: , <p>, <h1>, <ul>, <li>.

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 set size easily.
- Examples: , <a>, , , <img>.

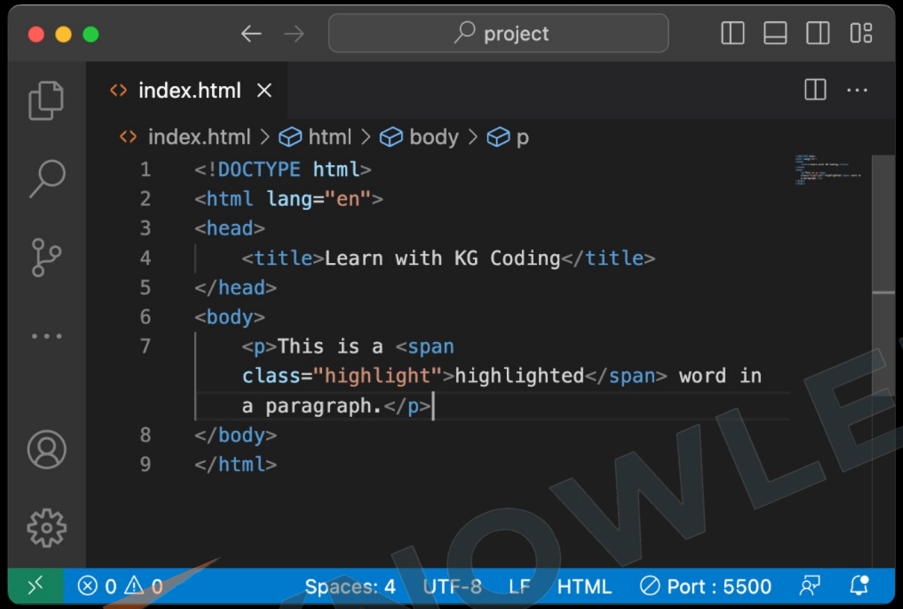
4.3 Div Tags

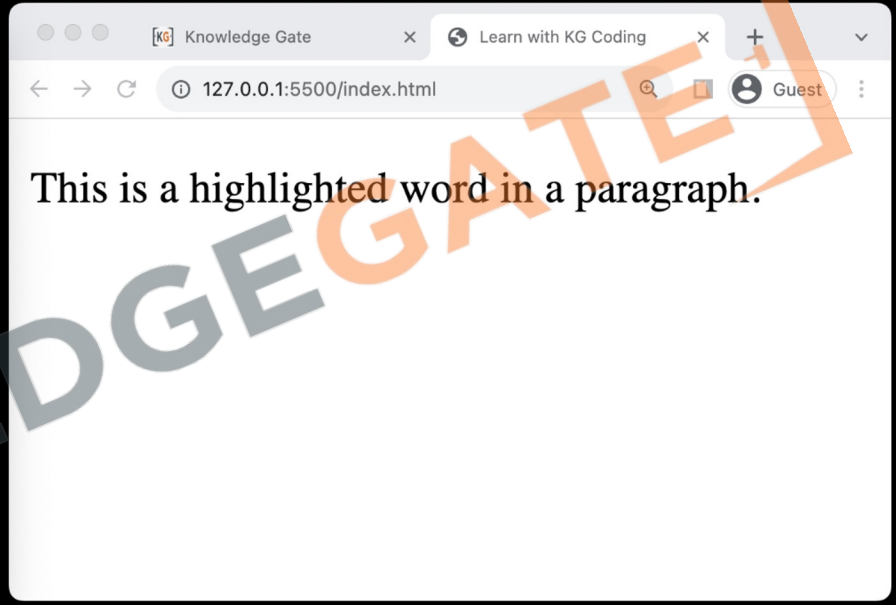




1. Purpose: Acts as a container for other HTML elements.
2. Non-Semantic: Doesn't provide inherent meaning to enclosed content.
3. Styling: Commonly used for layout and styling via CSS.
4. Flexibility: Highly versatile and can be customized using classes or IDs.

4.4 Span Tags



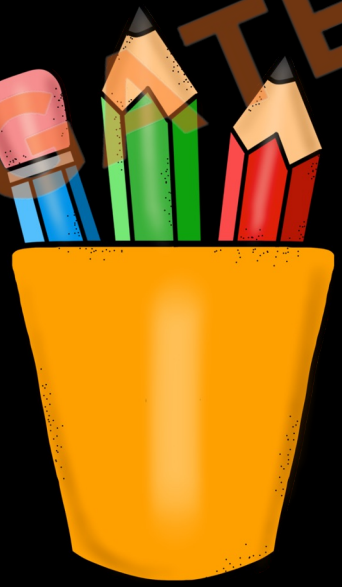


1. Purpose: Used for inline elements to style or manipulate a portion of text.
2. Non-Semantic: Doesn't add specific meaning to the enclosed text.
3. Styling: Commonly used for changing color, font, or adding effects via CSS.
4. Inline Nature: Doesn't break text flow or create a new block-level element.

Level 4 Revision

HTML and Project Structure

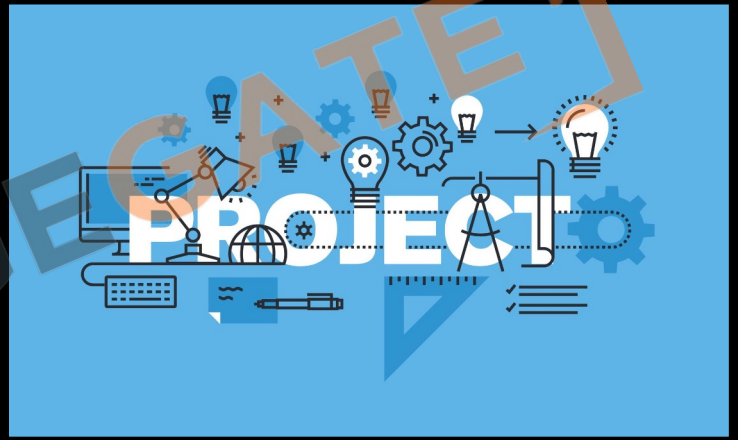
1. Semantic Tags
    1. Semantic / Non-Semantic Tags
2. Body Tags
    1. Header Tag
    2. Main Tag
        1. Section Tag
        2. Article Tag
        3. Aside Tag
    3. Footer Tag
3. Folder Structure
    1. Recommended Folder structure
4. More Tags
    1. Navigation tags
    2. Block / Inline Elements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



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.



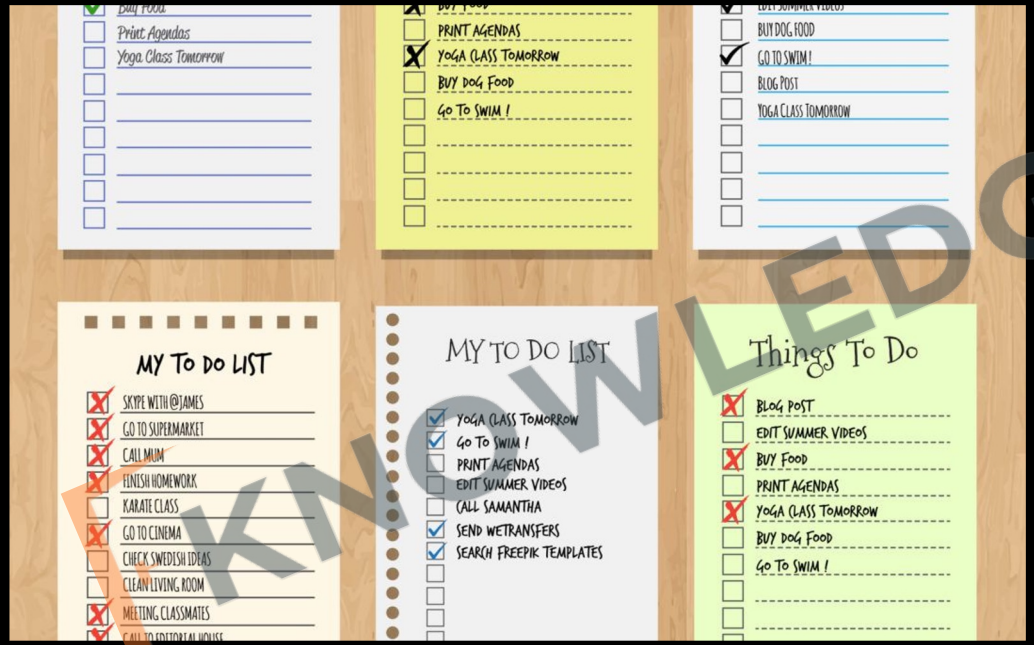
Level 5

List, Tables & Forms

1. List Tag
    1. Ordered Lists
    2. Types of Ordered Lists
    3. Unordered Lists
2. Table Tag
    1. <tr>, <td>, <th> tags
    2. Captions
    3. Col spans
3. Forms
    1. Input tag
    2. Action Attributes
    3. Name and Value Property
    4. Label Tag
    5. Exploring Types
4. iFrame Tag
    1. Using iFrames

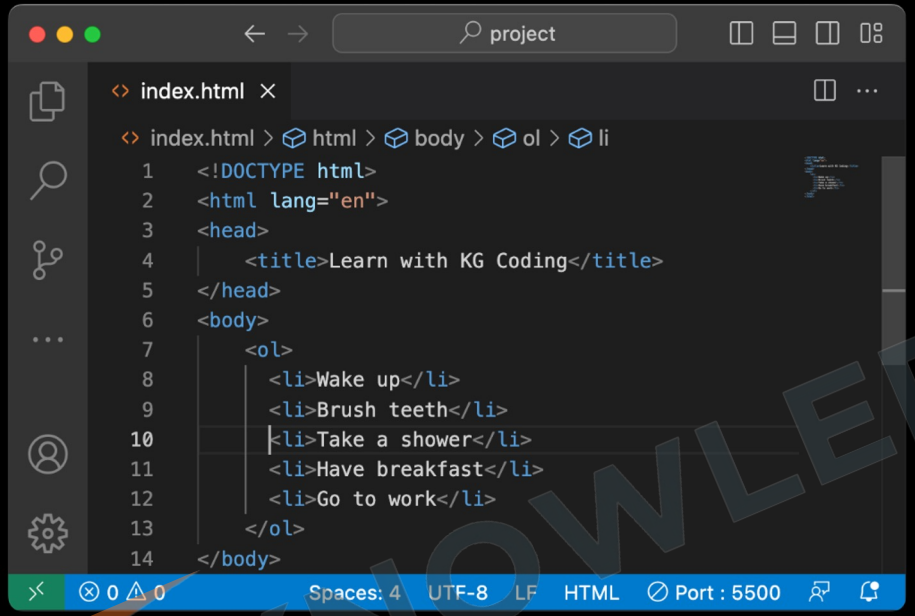
Level 5

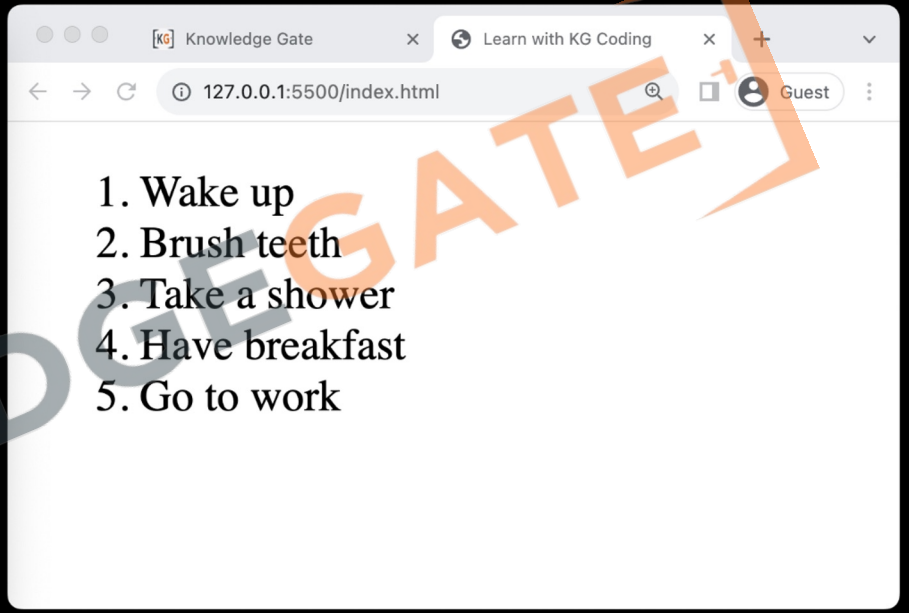
List, Tables & Forms



1. List Tag

1.1 Ordered Lists



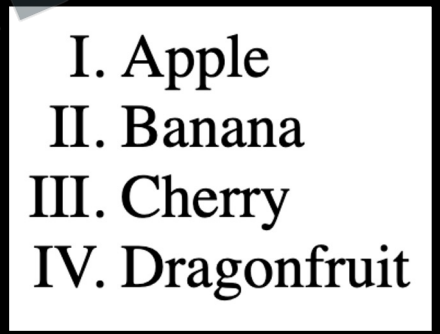


1. Purpose: Used for creating lists with items that have a specific order.
2. Default: Items are automatically numbered.
3. Nesting: Can be nested within other lists.

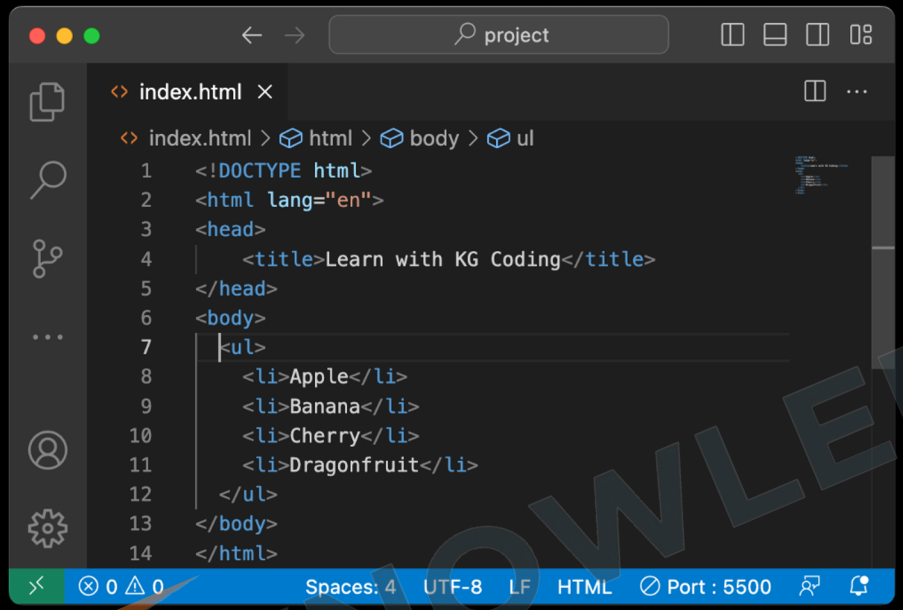
1.2 Types of Ordered Lists

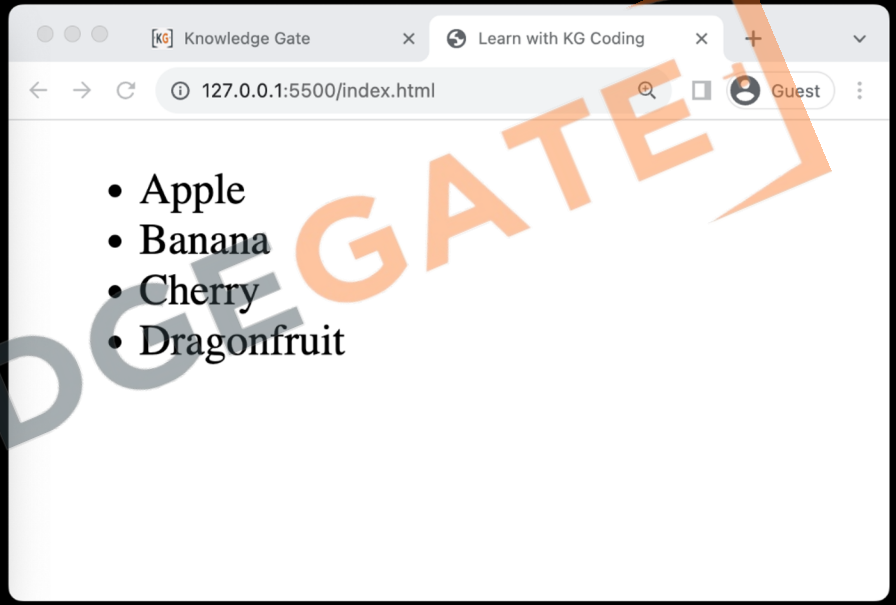
Ordered Lists

- 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"



1.3 Unordered Lists





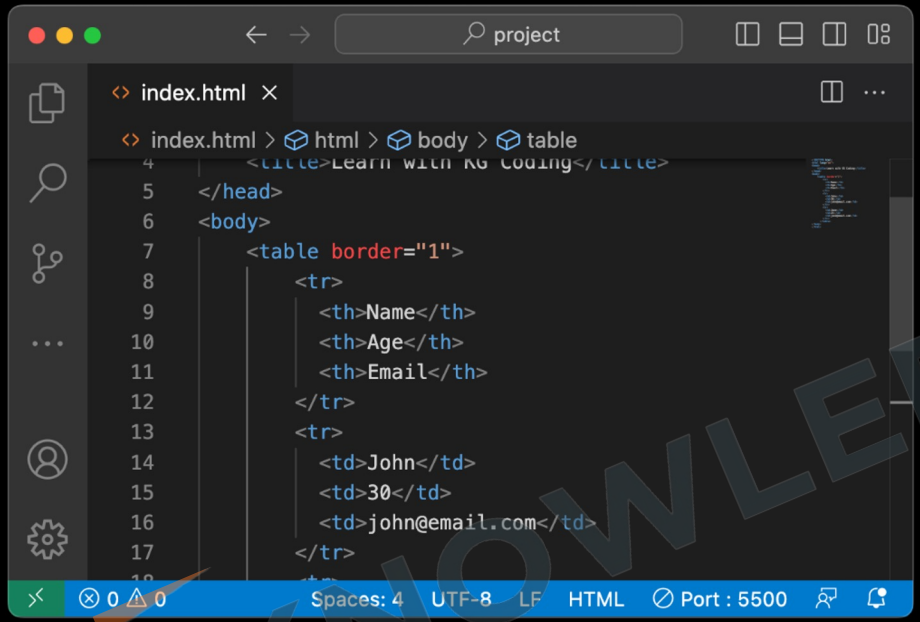
1. Purpose: Used for lists where the order of items doesn't matter.
2. Default: Items are usually bulleted.
3. Nesting: Can be nested within other lists.

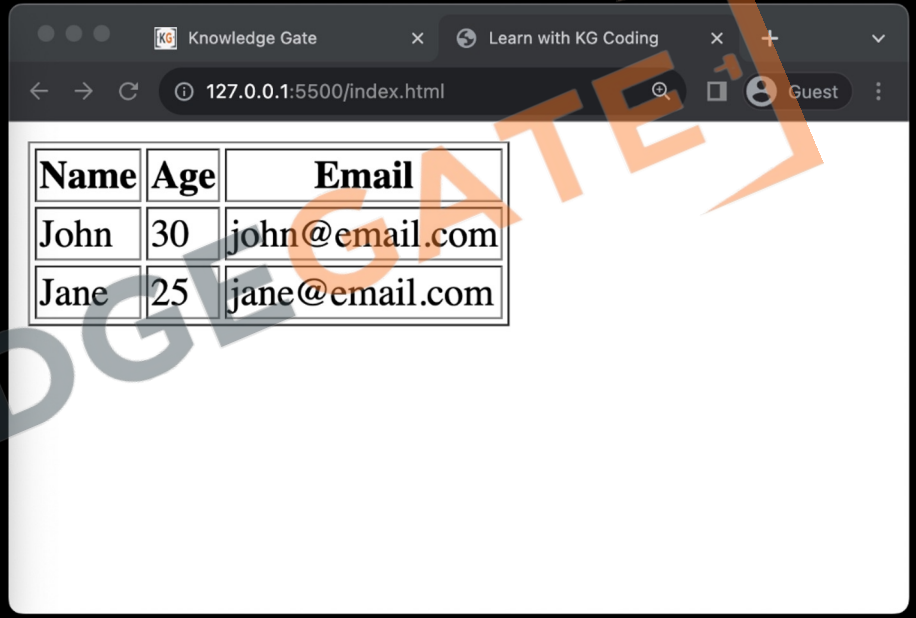
Level 5

List, Tables & Forms

2. Table Tag

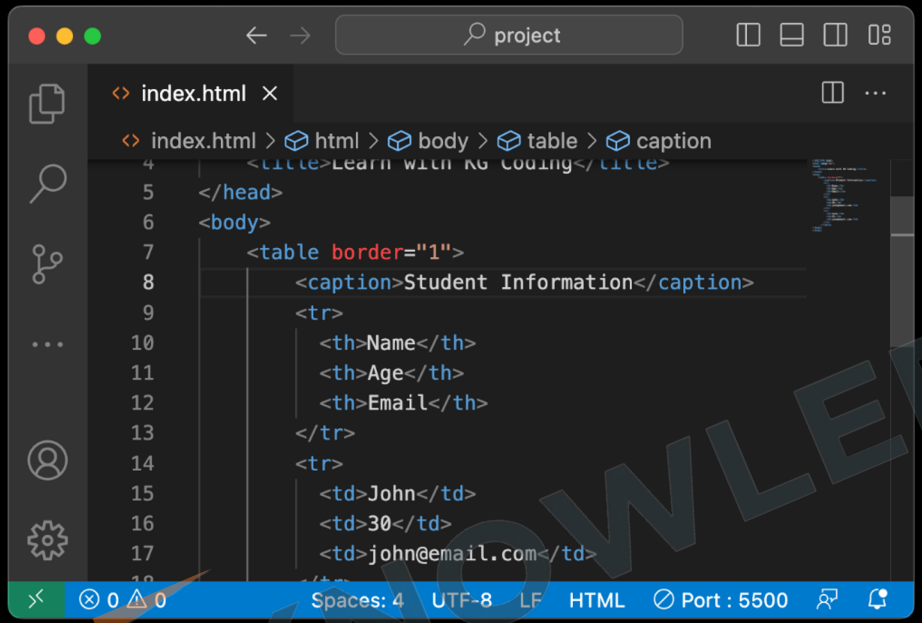
2.1 <tr>, <td>, <th> Tags

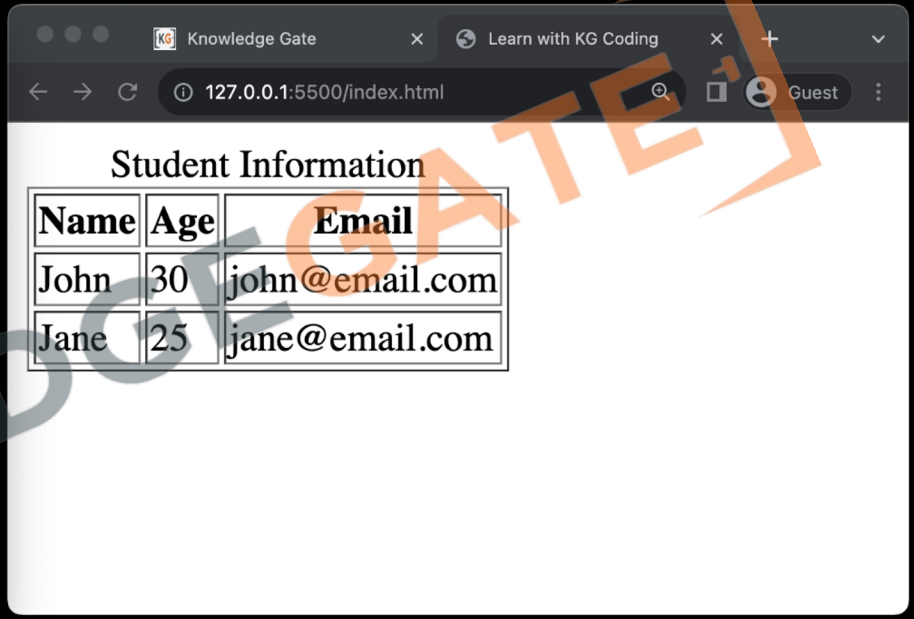




1. <tr> Table Row : Used to define a row in an HTML table.
2. <th> Table Header : Used for header cells within a row.
   Text is bold and centered by default.
3. <td> Table Data : This Holds the actual data.

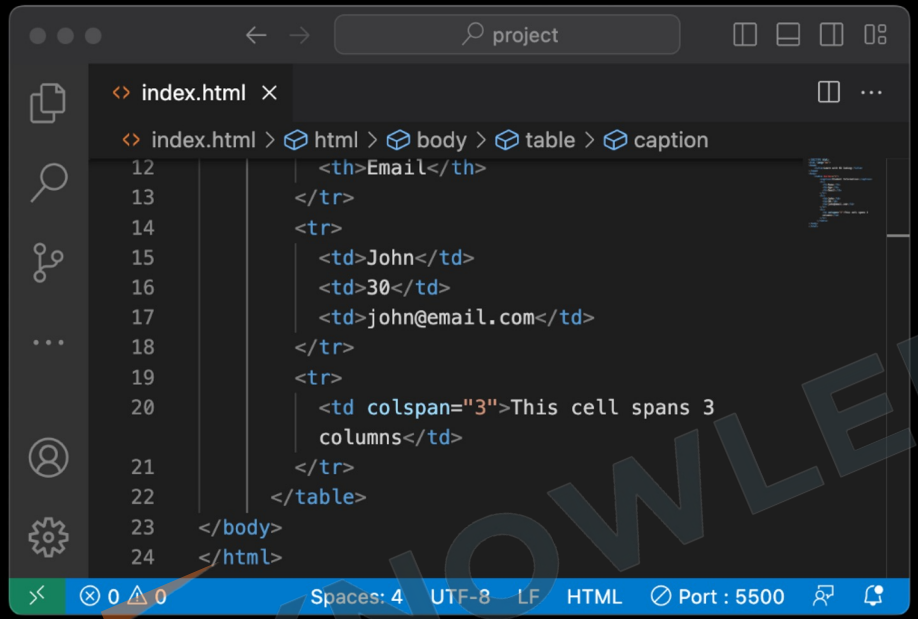
2.2 Captions

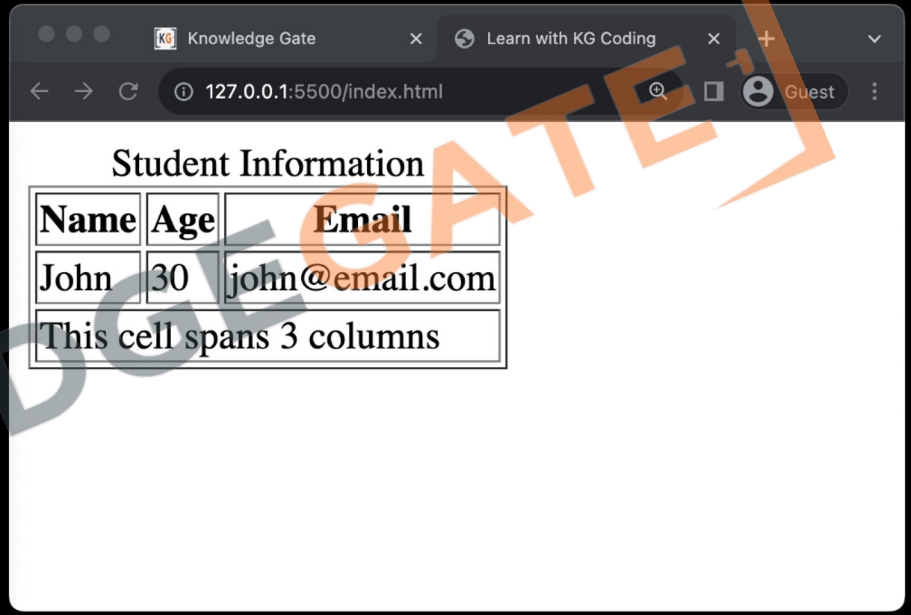




1. Purpose: Provides a title or description for a table.
2. Placement: Must be inserted immediately after the <table> opening tag.
3. Alignment: Centered above the table by default.
4. Accessibility: Helps screen readers understand the table's purpose.

2.3 Col Spans





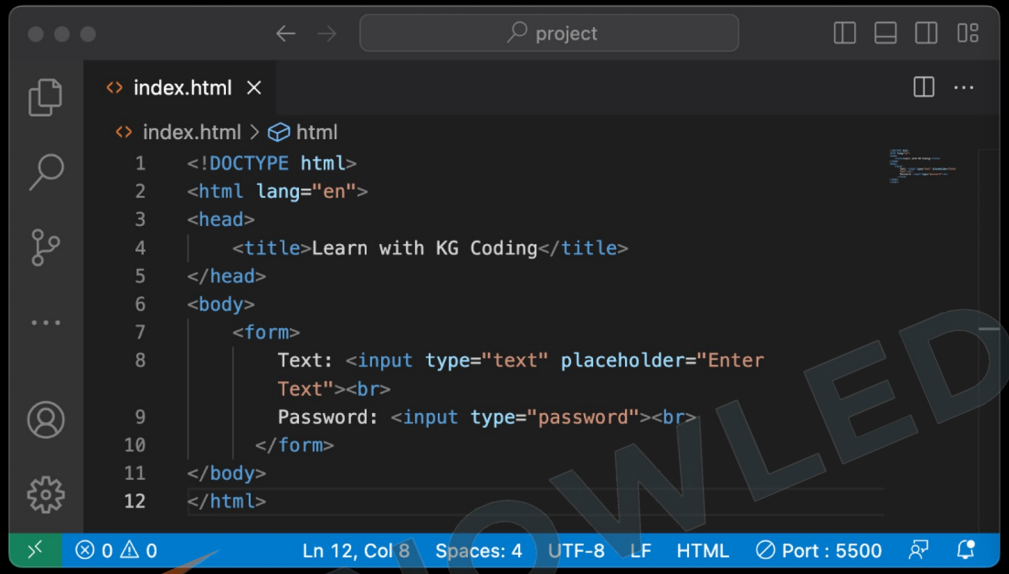
1. Attribute: Uses the colspan attribute in <td> or <th> tags.
2. Purpose: Allows a cell to span multiple columns horizontally.
3. Alignment: Takes the space of the specified number of columns.
4. Layout: Useful for combining cells to create complex table layouts.

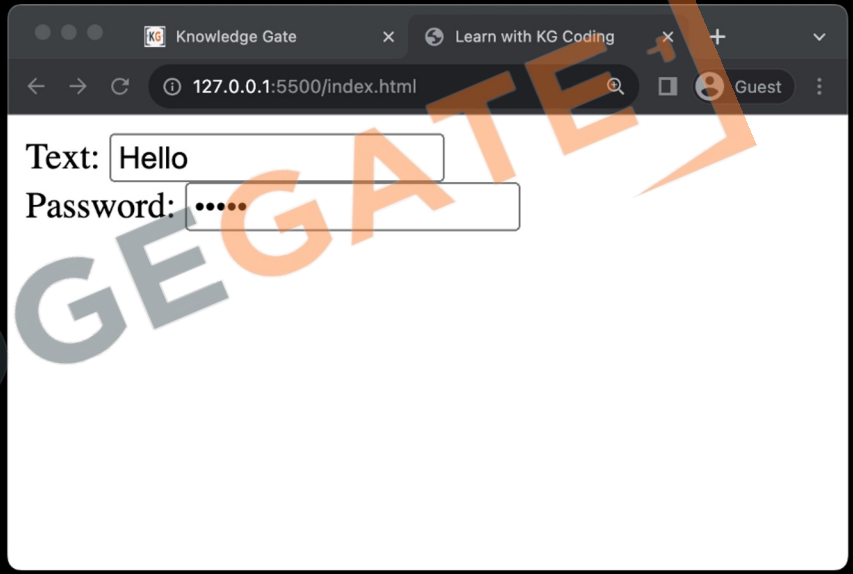
Level 5

List, Tables & Forms

3. Forms

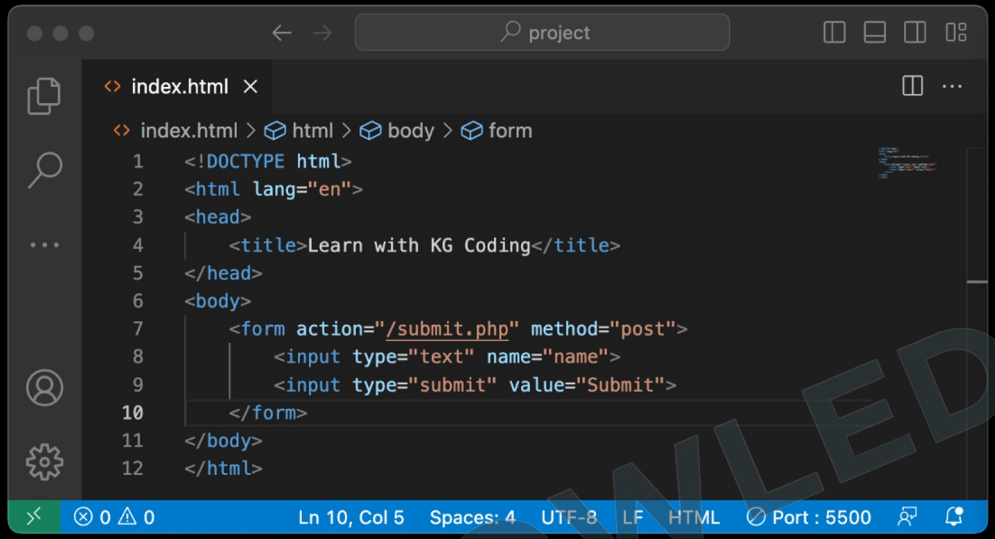
3.1 Input Tag

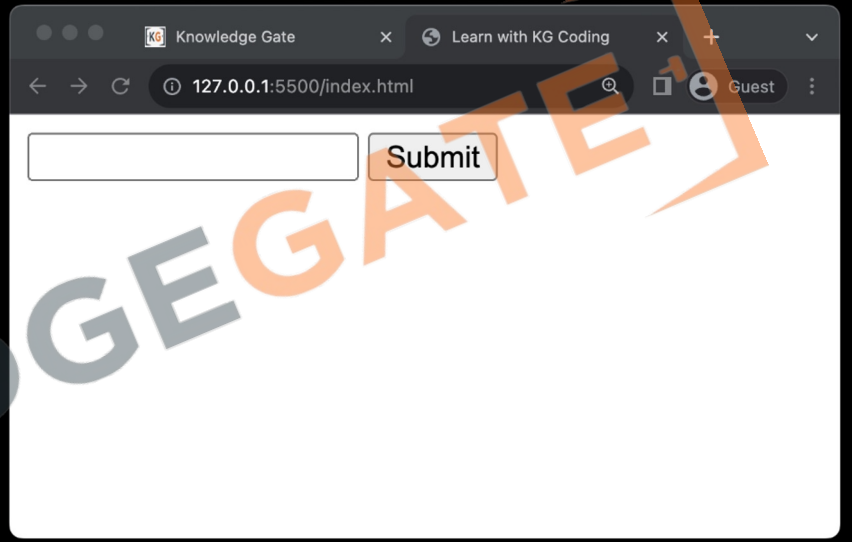




1. Purpose: Used within a <form> element to collect user input.
2. Self-Closing: The <input> tag is self-closing; doesn't require a closing tag.
3. Attributes: Common attributes are name, value, placeholder, and required.

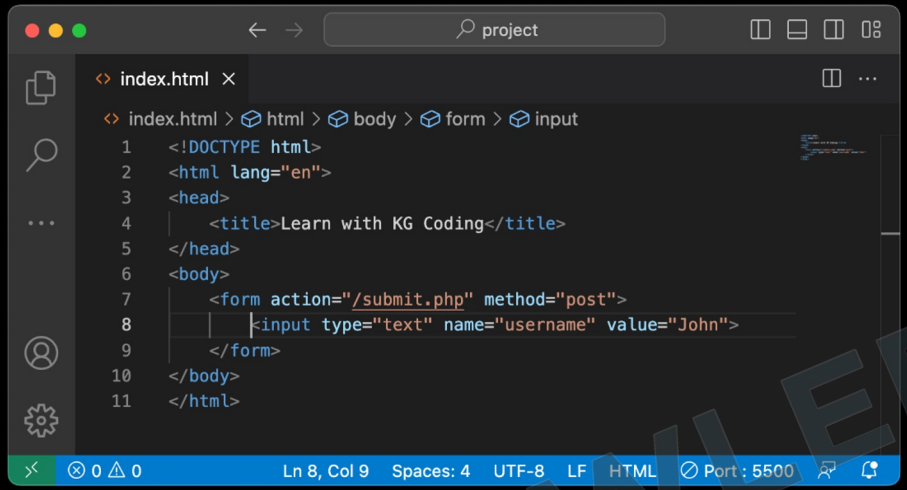
3.2 Action attribute

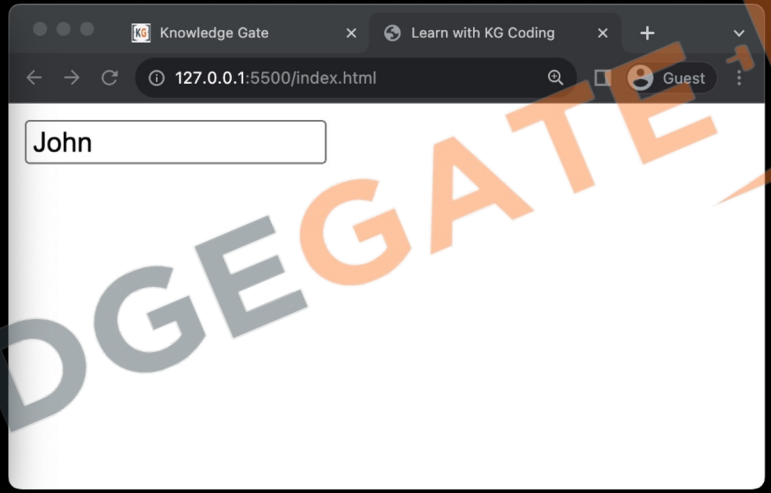




1. Purpose: Specifies the URL to which the form data should be sent when submitted.
2. Default: If not specified, the form will be submitted to the current page's URL.
3. Server-Side: Usually points to a server-side script (like PHP, Python, etc.) that processes the form data.

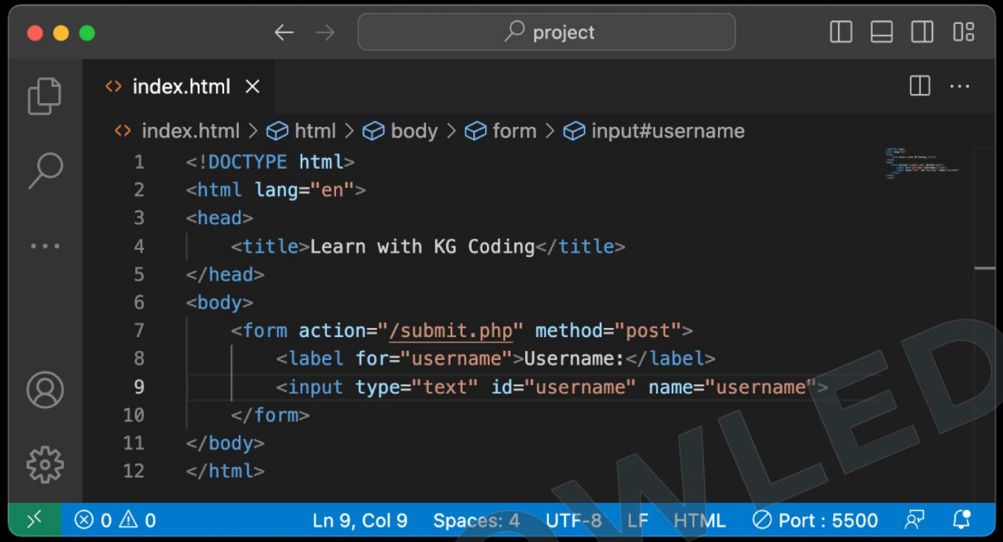
3.3 Name and Value property

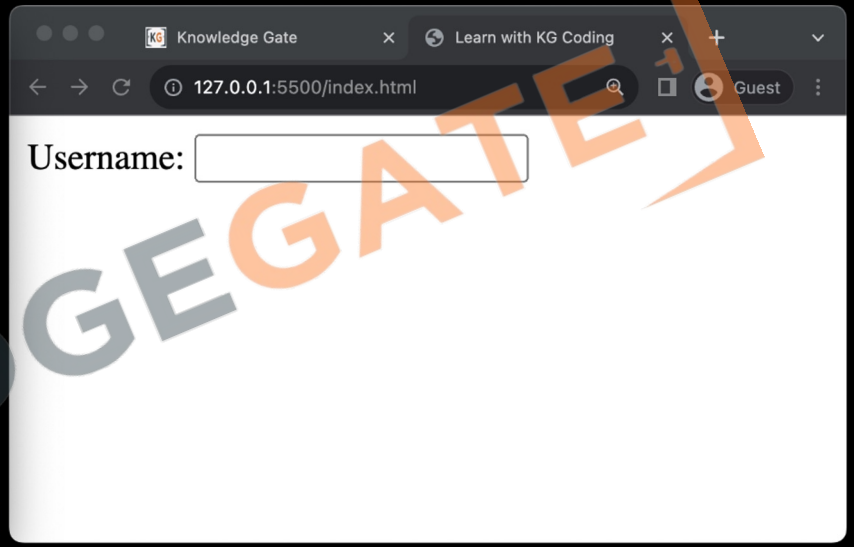




- `name` Property:
  - ID for Data: Identifies form elements when submitting.
  - Unique: Should be unique to each element for clarity.
- `value` Property:
  - Default Data: Sets initial value for input elements.
  - Sent to Server: This is the data sent when form is submitted.

3.4 Label Tag



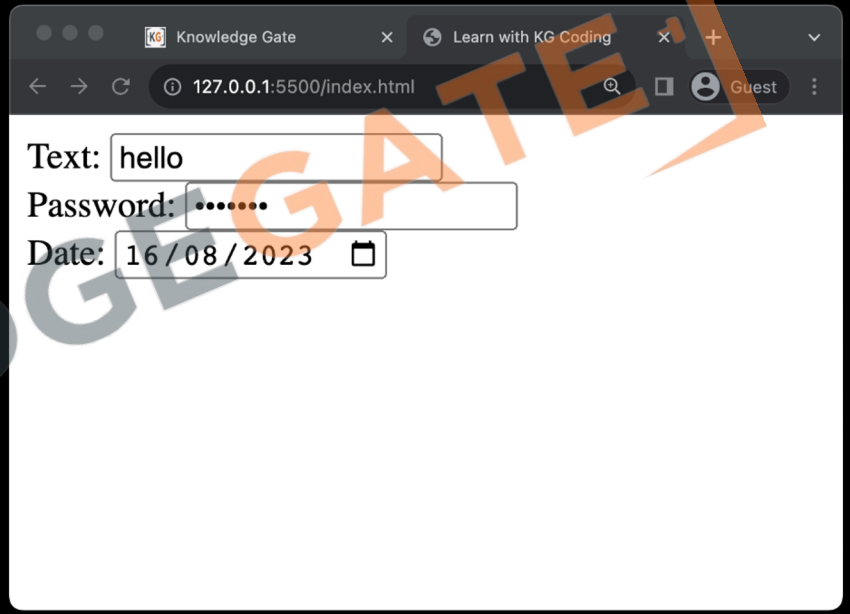


- Purpose: Adds a text description to form elements.
- for Attribute: Connects the label to a specific form element using the element's id.
- Accessibility: Makes the form more accessible.
- Readability: Enhances form readability and usability.

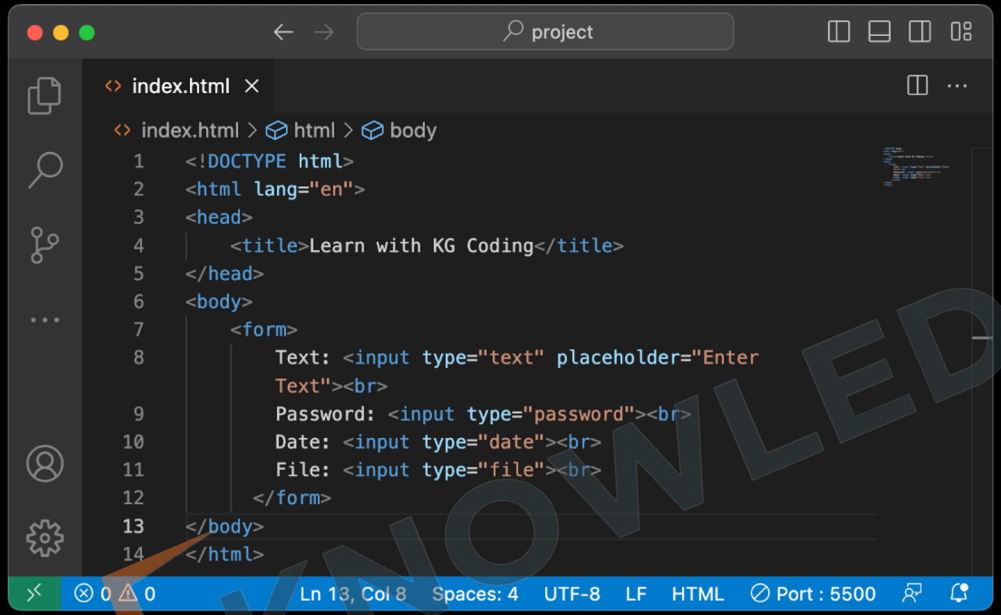
3.5 Input type: Date

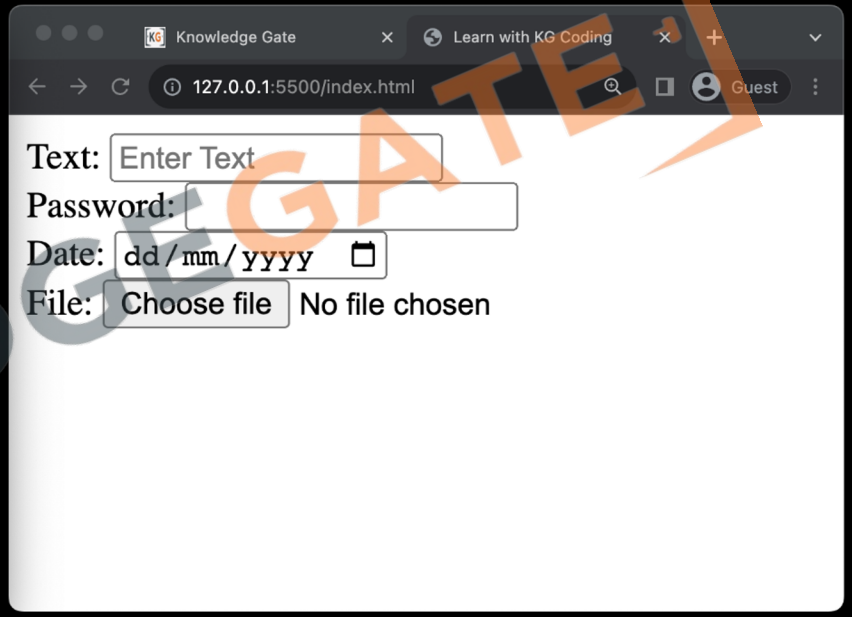
index.html

```
<!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>
        Date: <input type="date"><br>
    </form>
</body>
</html>
```

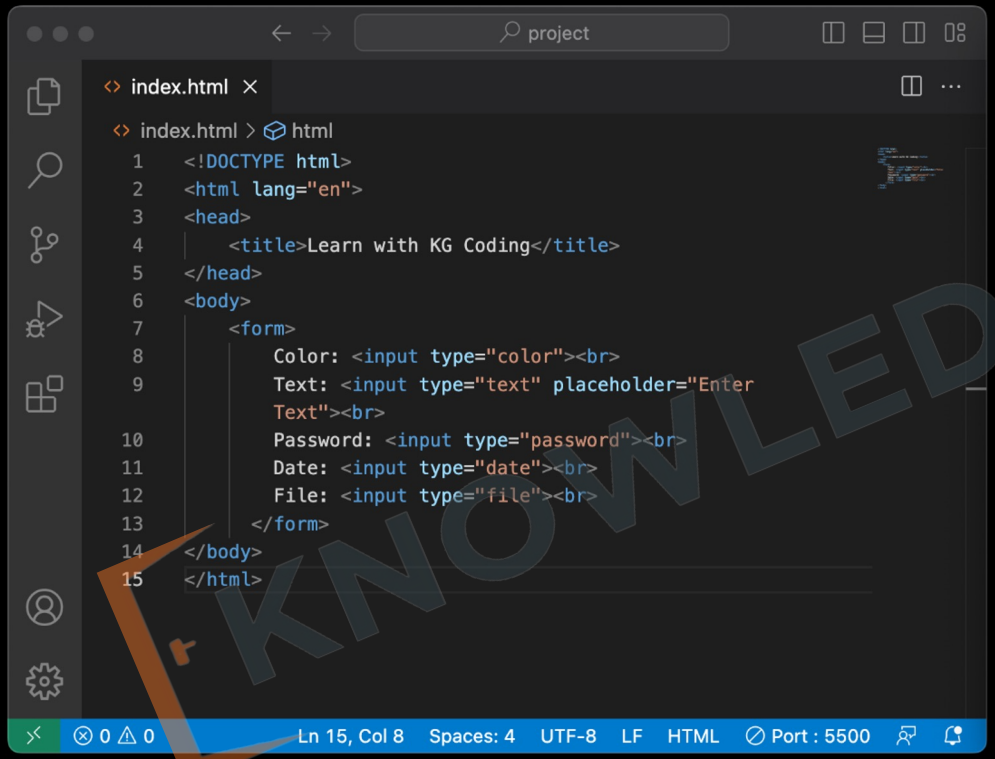


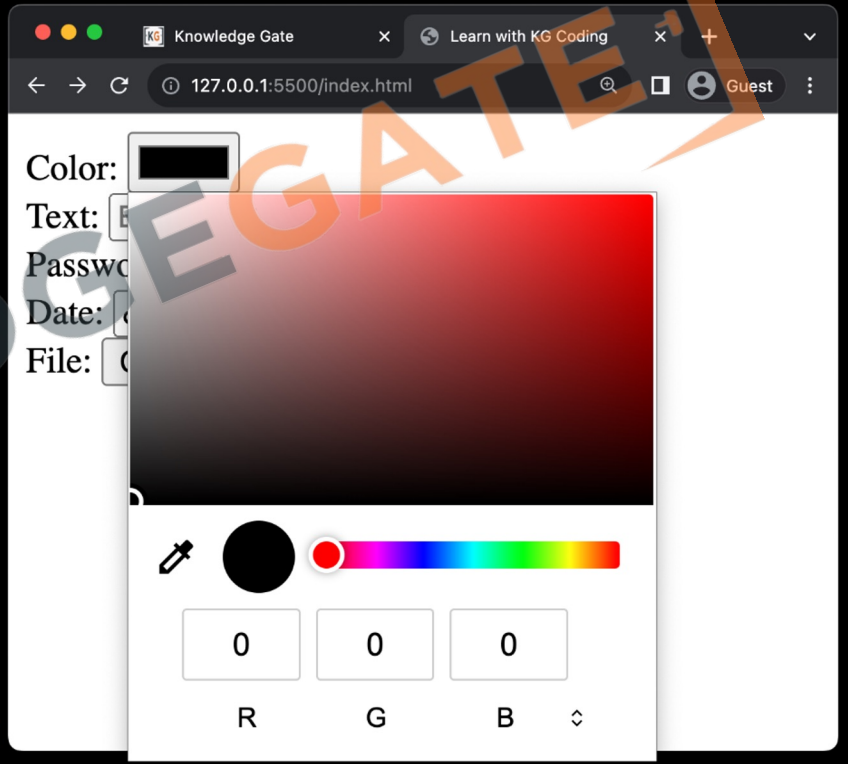
3.5 Input type: File



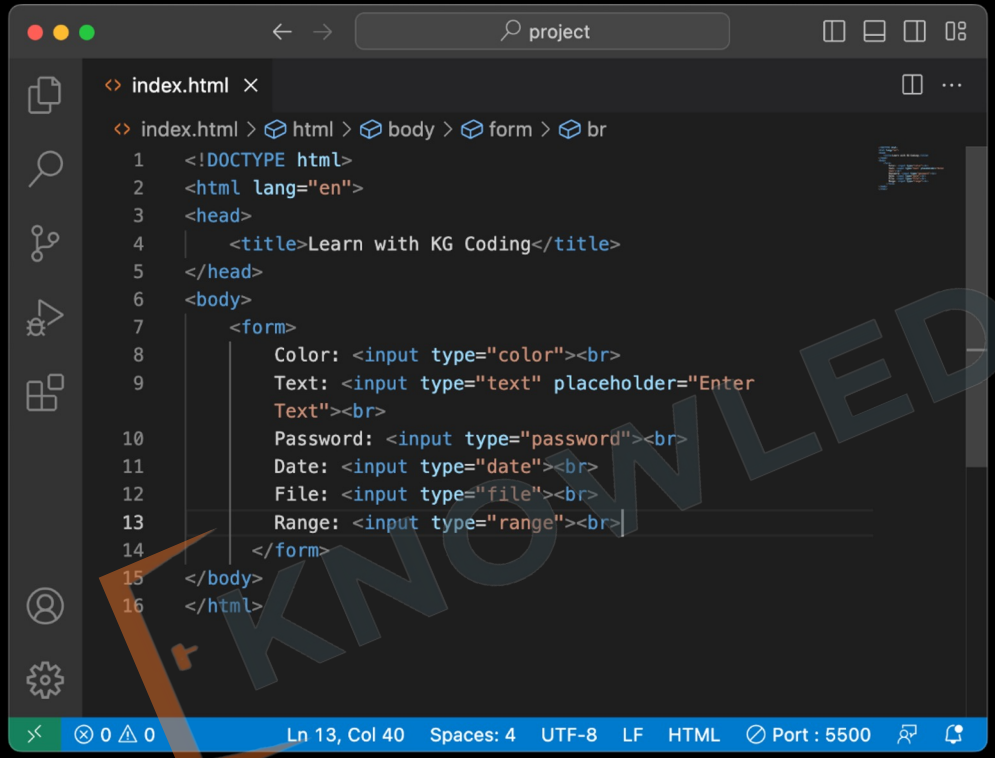


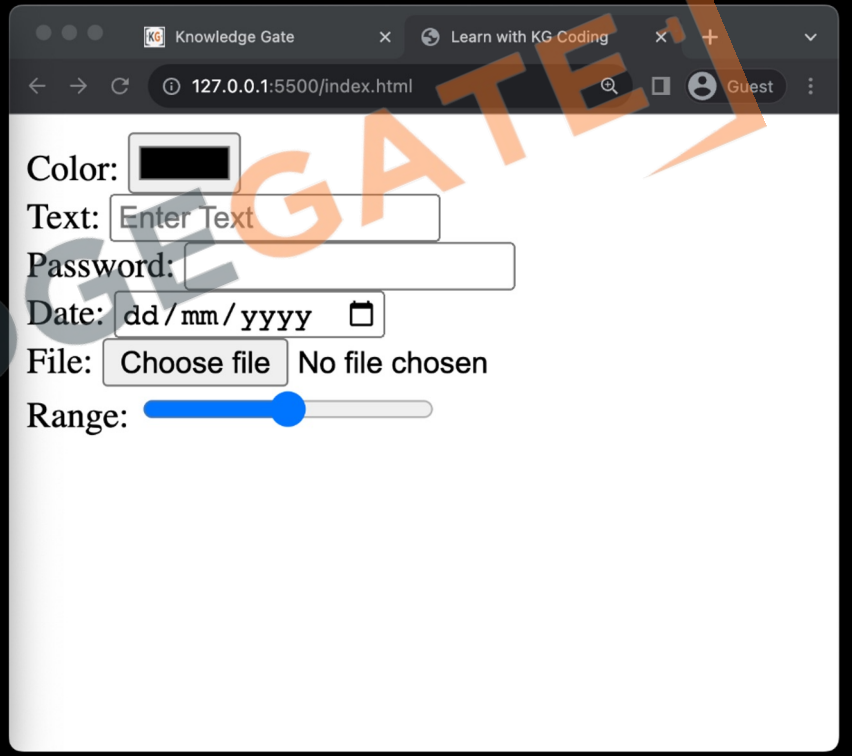
3.5 Input type: Color



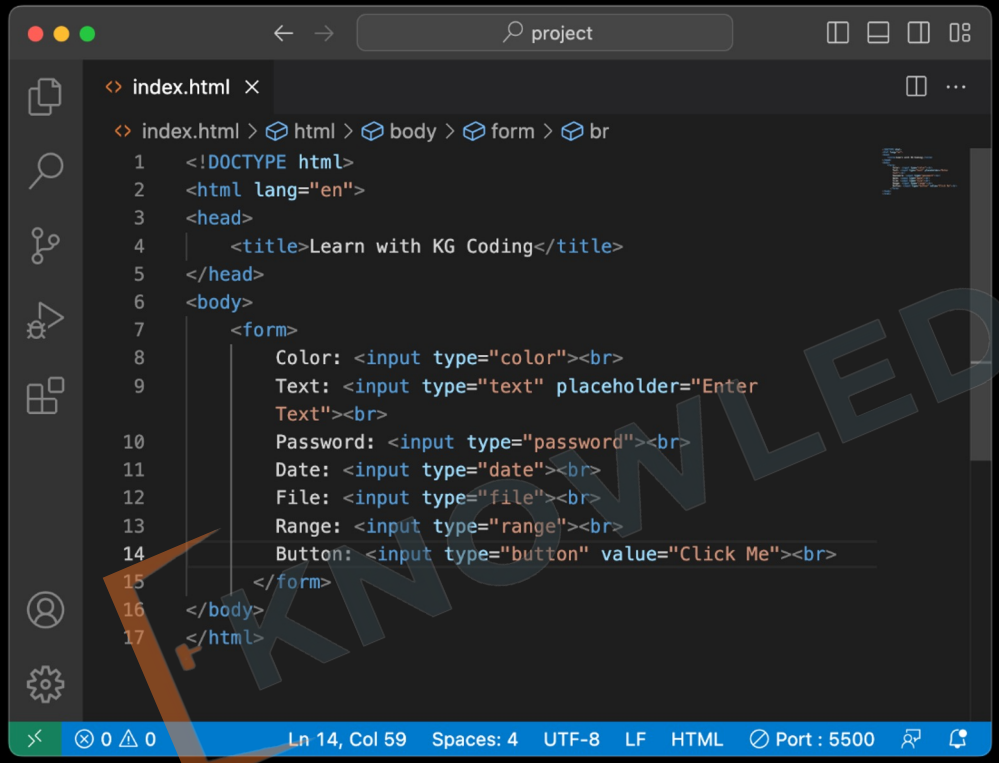


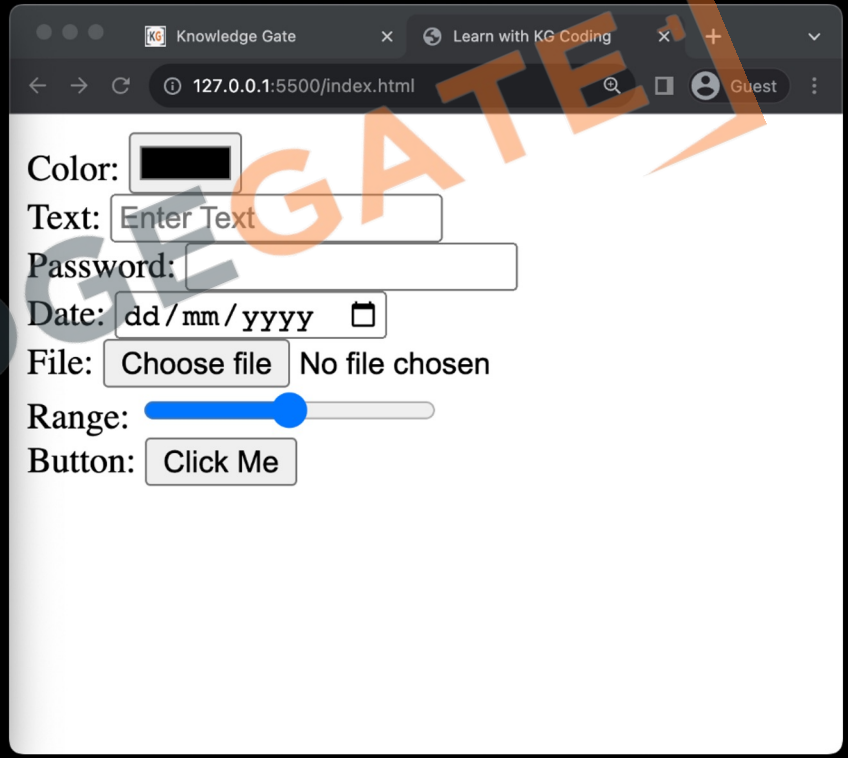
3.5 Input type: Range



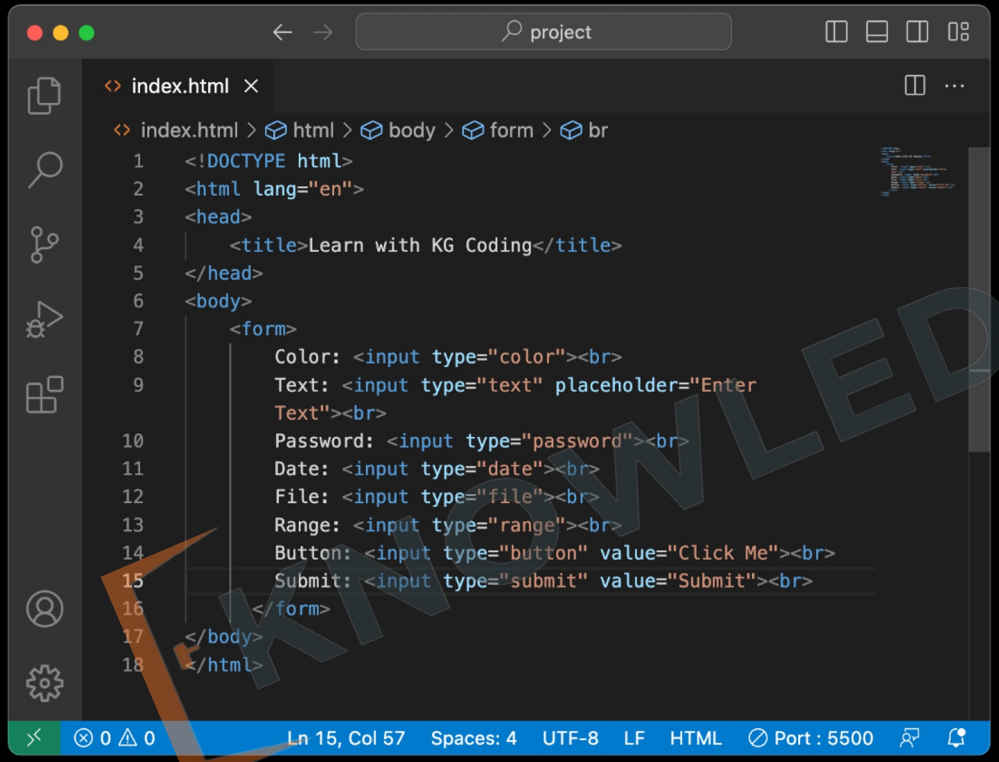


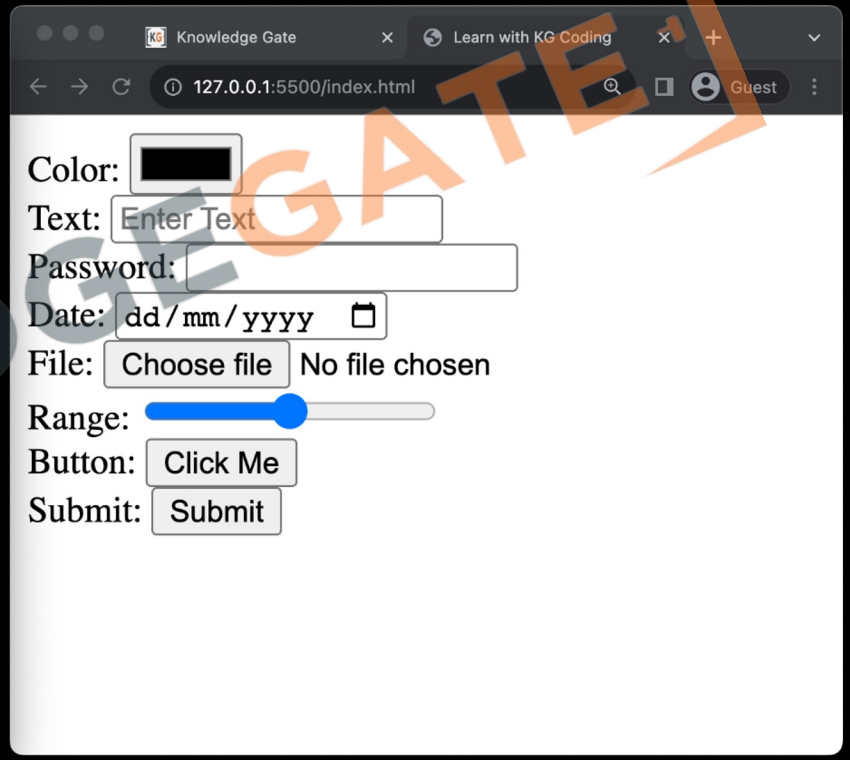
3.5 Input type: Button



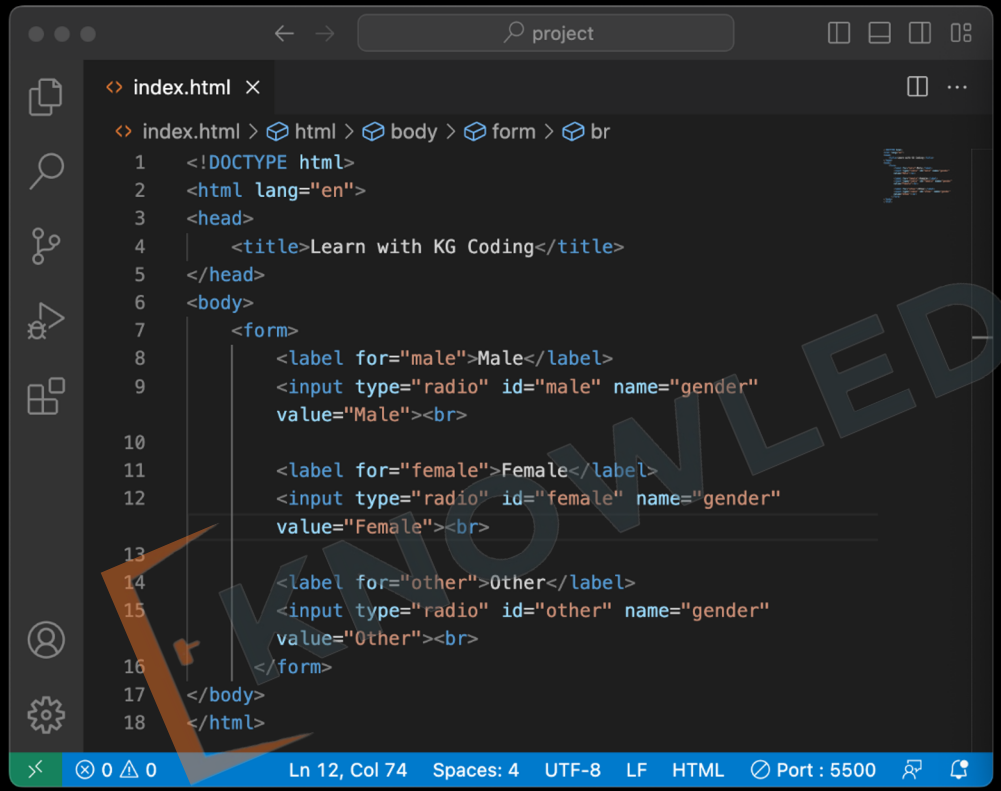


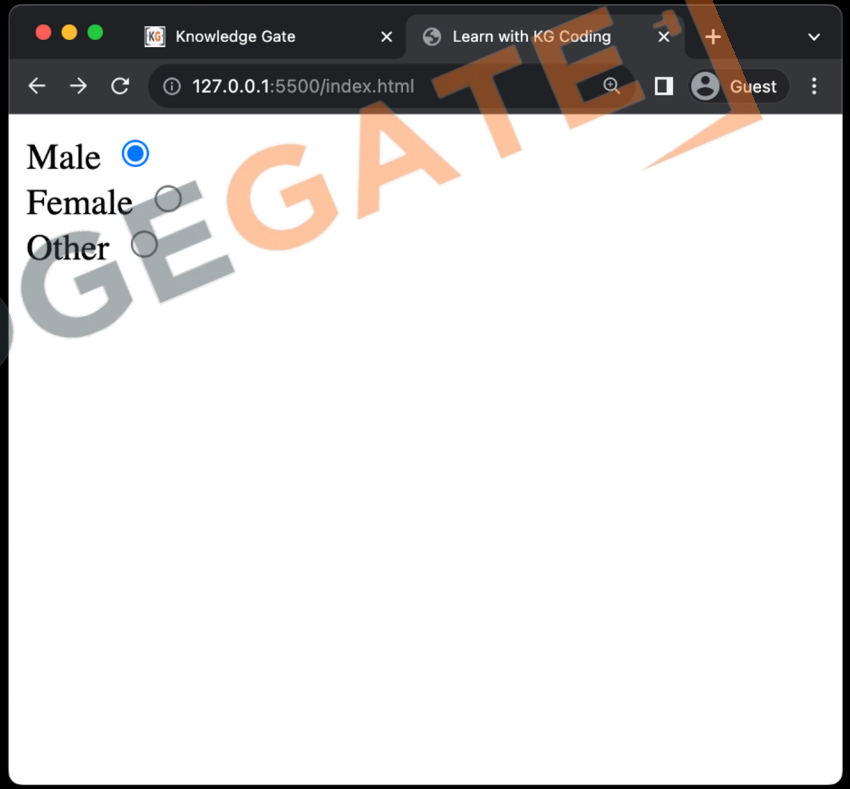
3.5 Input type: Submit



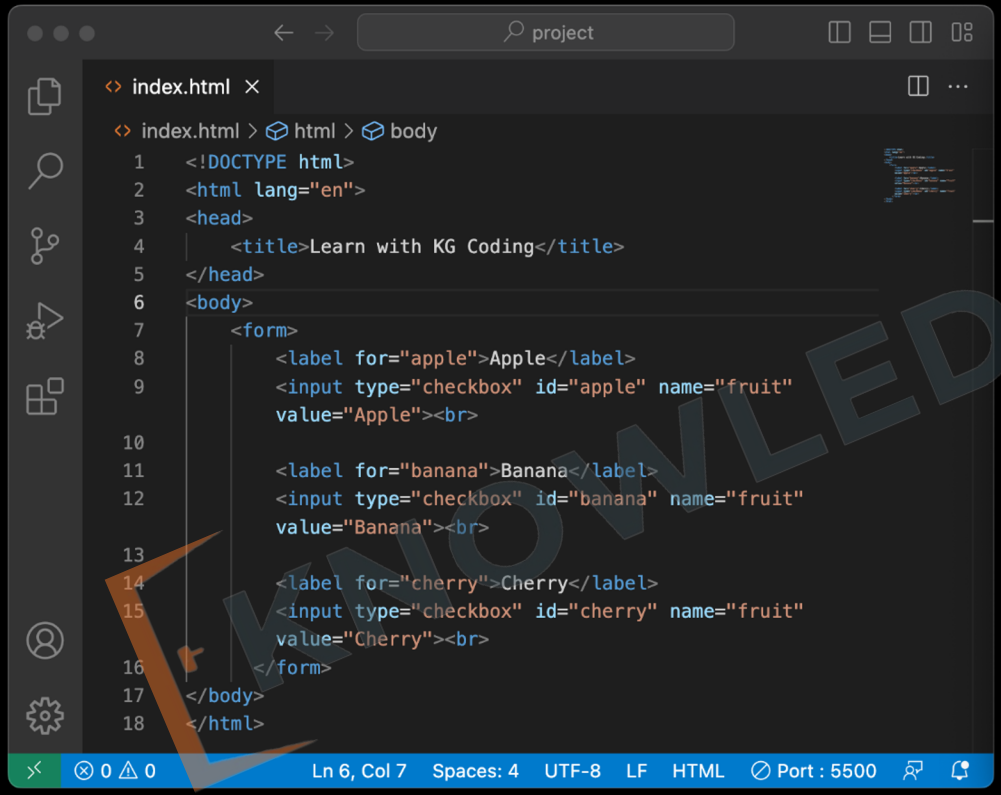


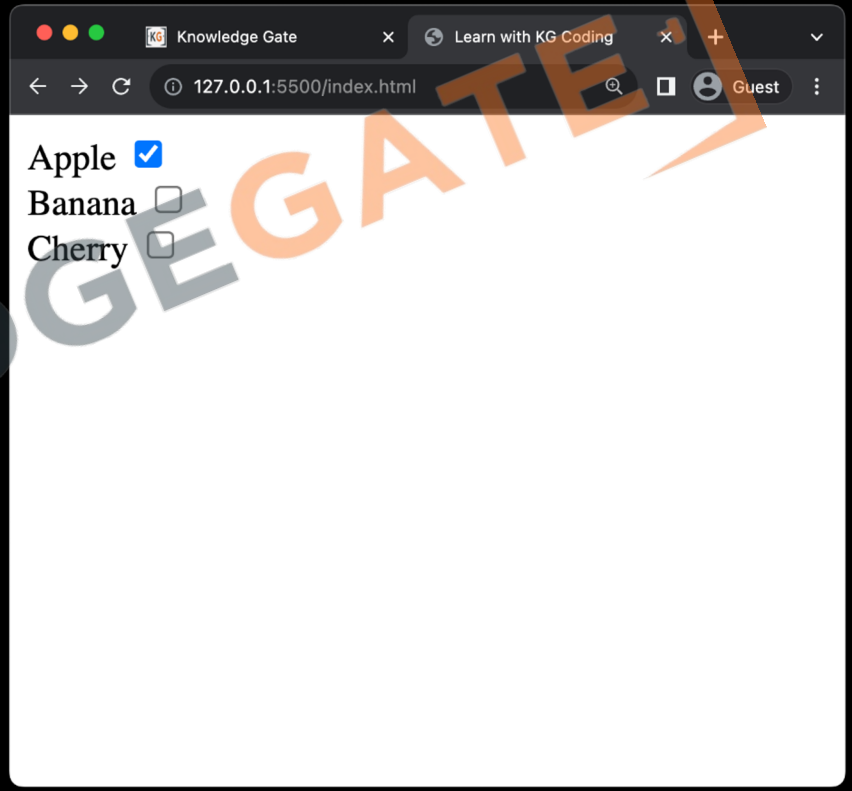
3.5 Input type: Radio



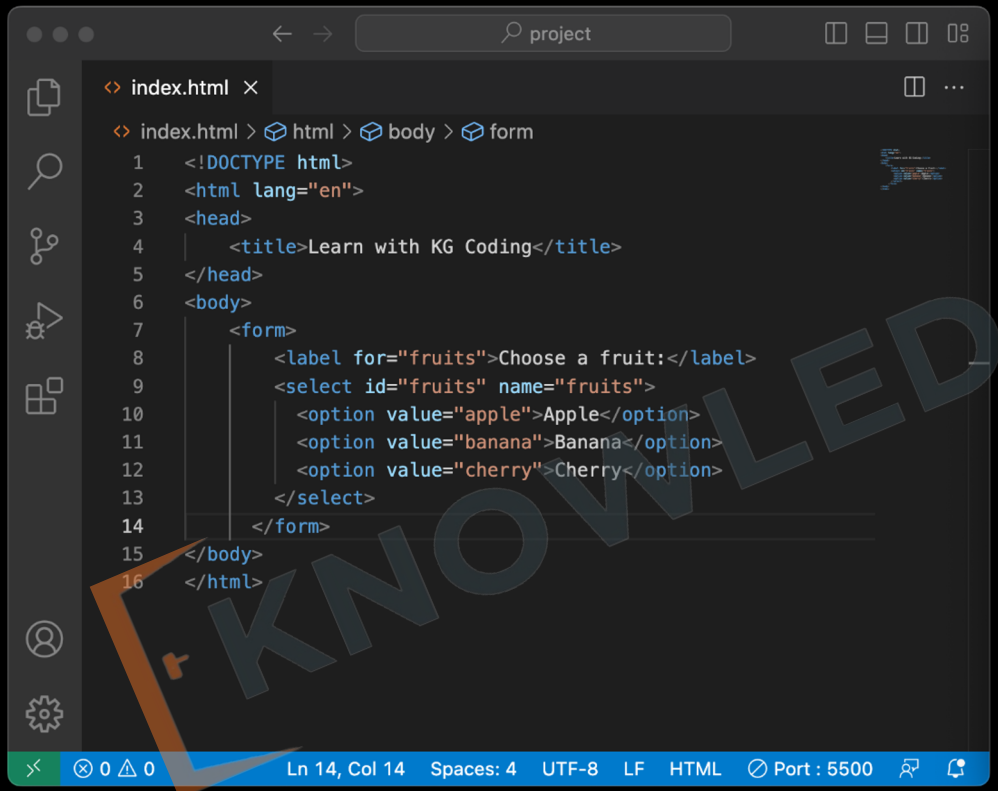


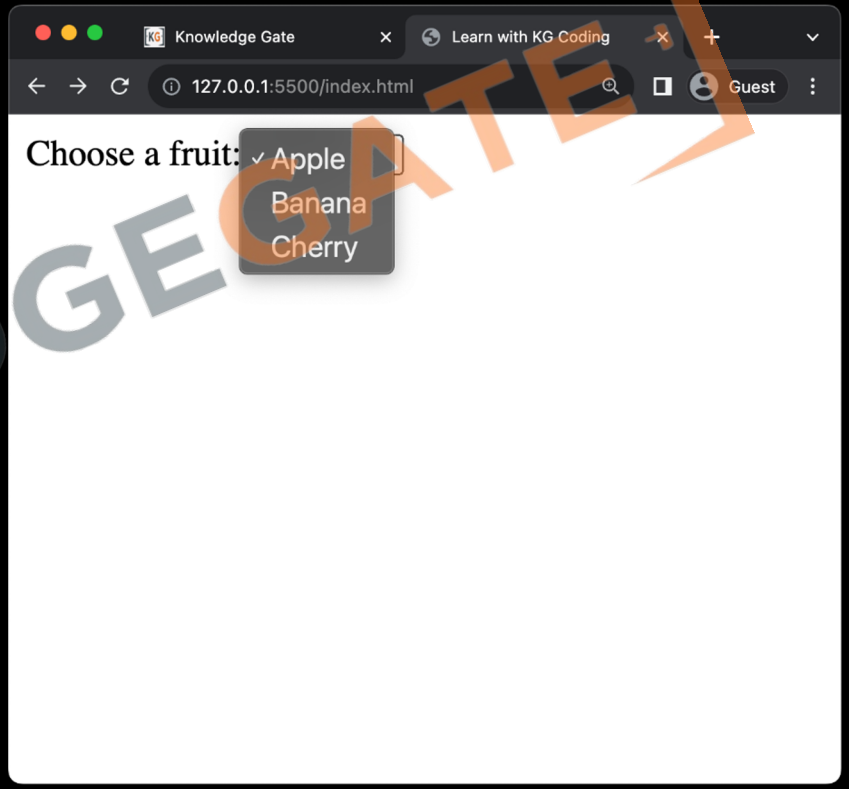
3.5 Input type: Checkbox



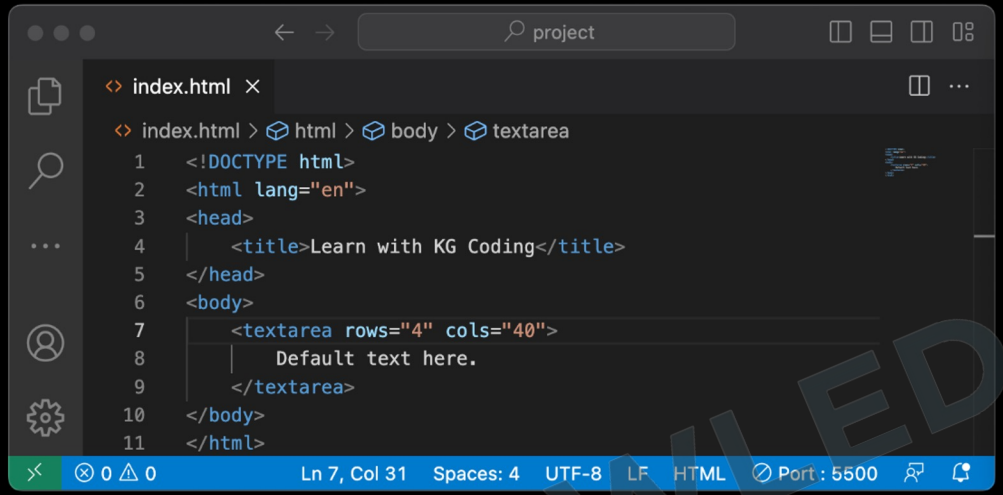


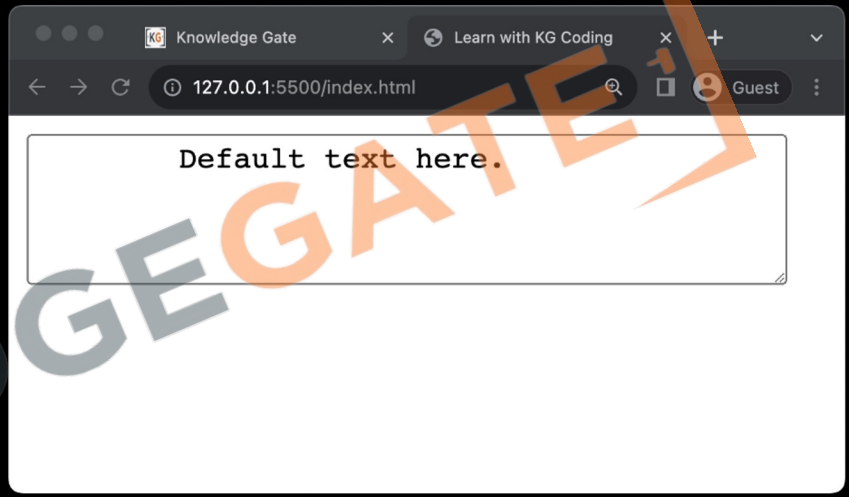
3.5 Input type: Select





3.5 Input type: TextArea





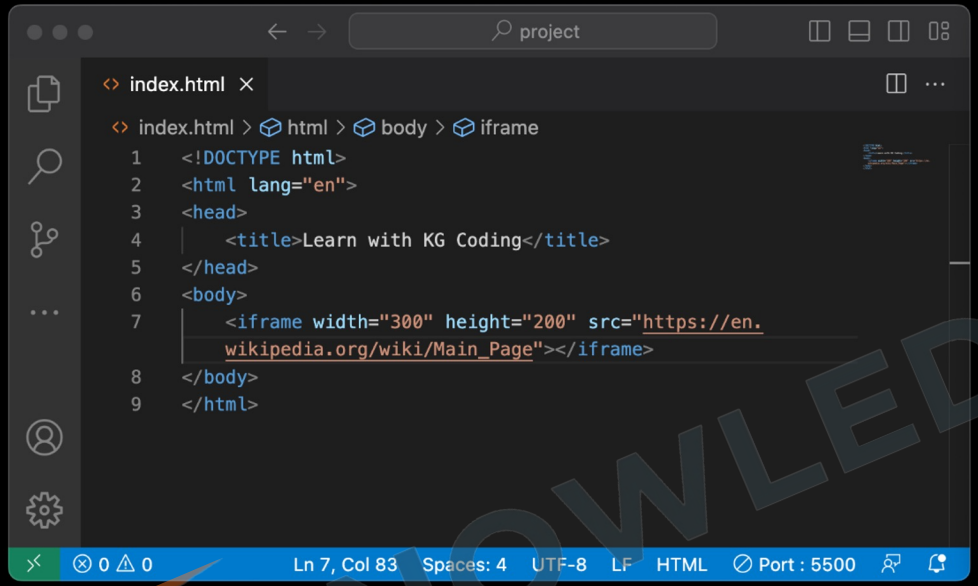
1. Purpose: <textarea> is used for multi-line text input in forms.
    1. rows Property: Specifies the visible number of lines in the textarea.
    2. cols Property: Sets the visible width measured in average character widths.
2. Resizable: Some browsers allow users to manually resize the textarea.

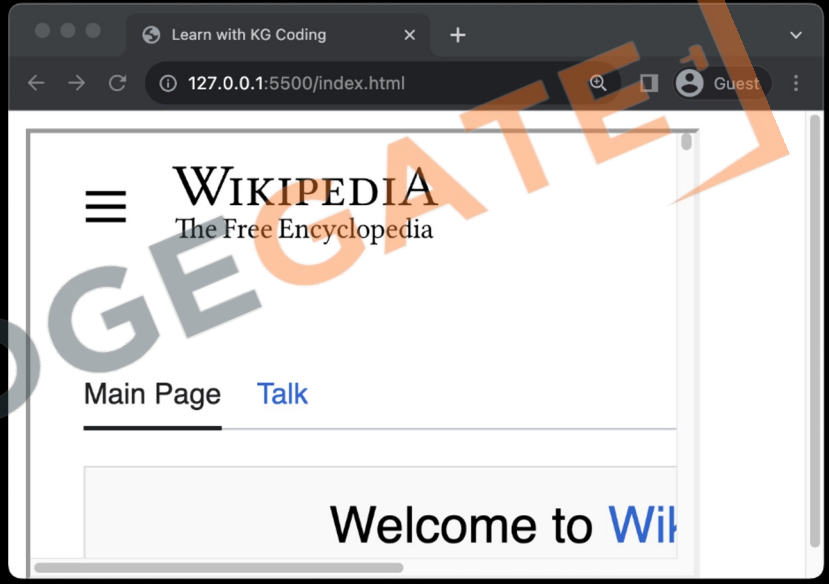
Level 5

List, Tables & Forms

4.iFrame Tag

4.1 Using iFrames





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.

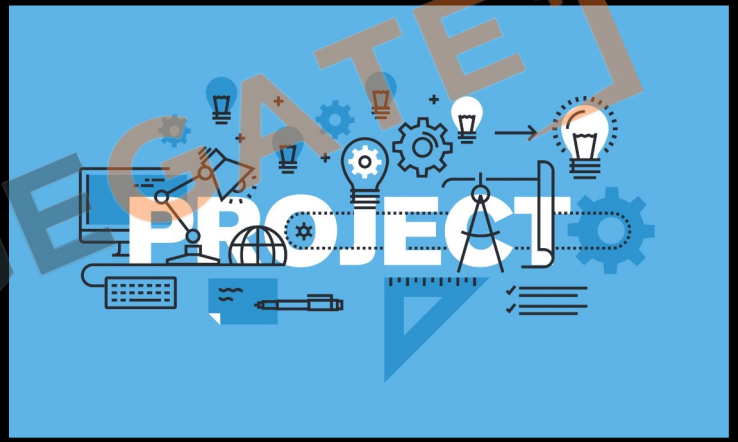
Level 5 Revision

List, Tables & Forms

1. List Tag
    1. Ordered Lists
    2. Types of Ordered Lists
    3. Unordered Lists
2. Table Tag
    1. <tr>, <td>, <th> tags
    2. Captions
    3. Col spans
3. Forms
    1. Input tag
    2. Action Attributes
    3. Name and Value Property
    4. Label Tag
    5. Exploring Types
4. iFrame Tag
    1. Using iFrames

Project Level 5

List, Tables & Forms

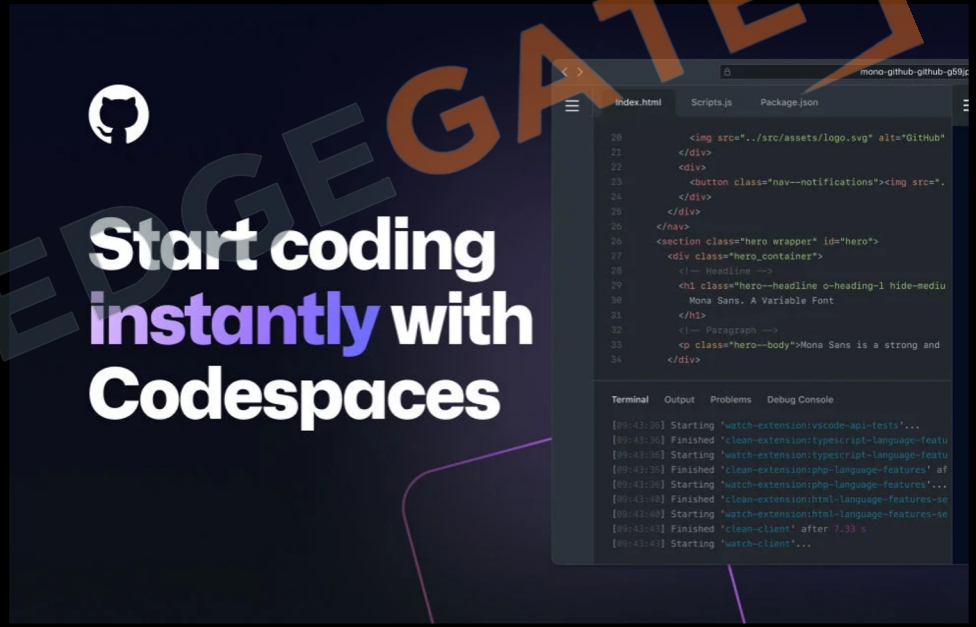


1. Create a page with all type of ordered list and one unordered list.
2. Create a table with headings, captions and a few rows. One of heading should take at least 3 columns.
3. Create a contact me form with relevant details for your resume website.
4. Use iFrame to add this video to your page.

Level Bonus

Github Pages & CodeSpace

1. Github
    1. What is Version Control
    2. What is Git and GitHub
    3. Account Creation
    4. Creating a Repo
    5. Creating a Codespace
    6. Creating a Github page
    7. Publishing our project
2. FrameWorks
    1. React
    2. Angular
    3. Vue



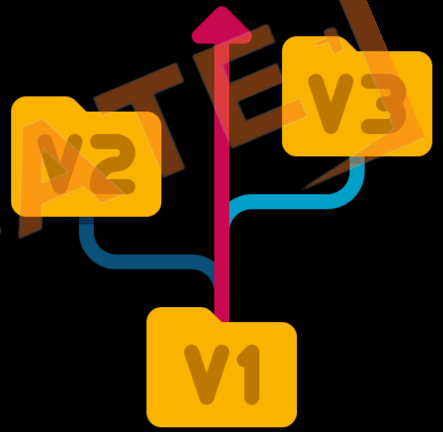
Level Bonus

Github Pages & CodeSpace

1. GitHub

1.1 What is Version Control

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 versions.



1.2 What is Git and GitHub

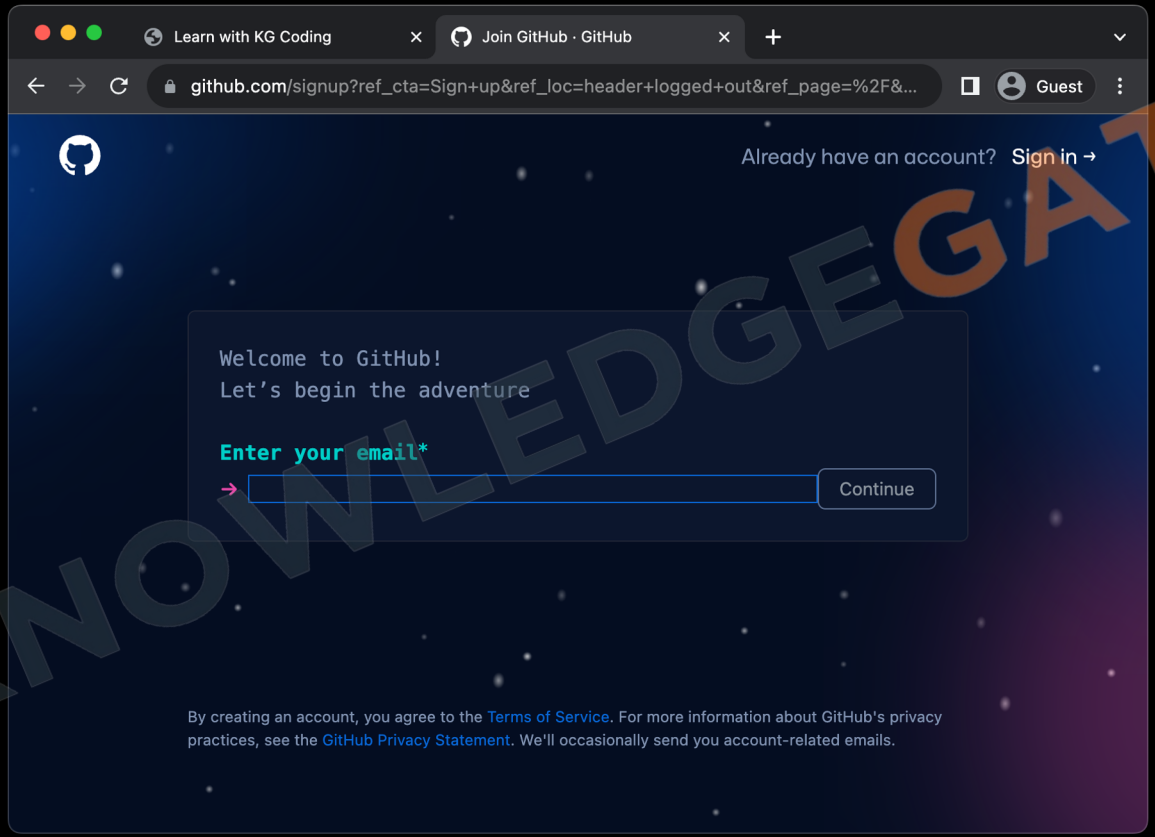
What is Git?

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

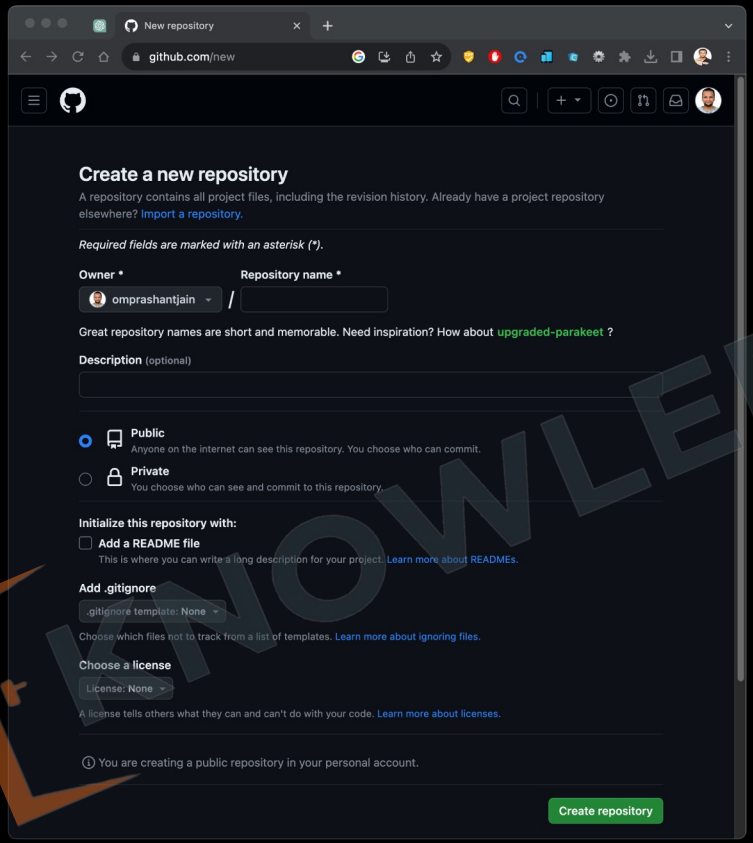
What is GitHub?

- Definition: A web service for hosting and collaborating on Git repositories.
- Fork: Creates a personal copy of another user's repository.
- Pull Request: A way to propose changes to existing code.
- Issues: Used for tracking bugs and feature ideas.

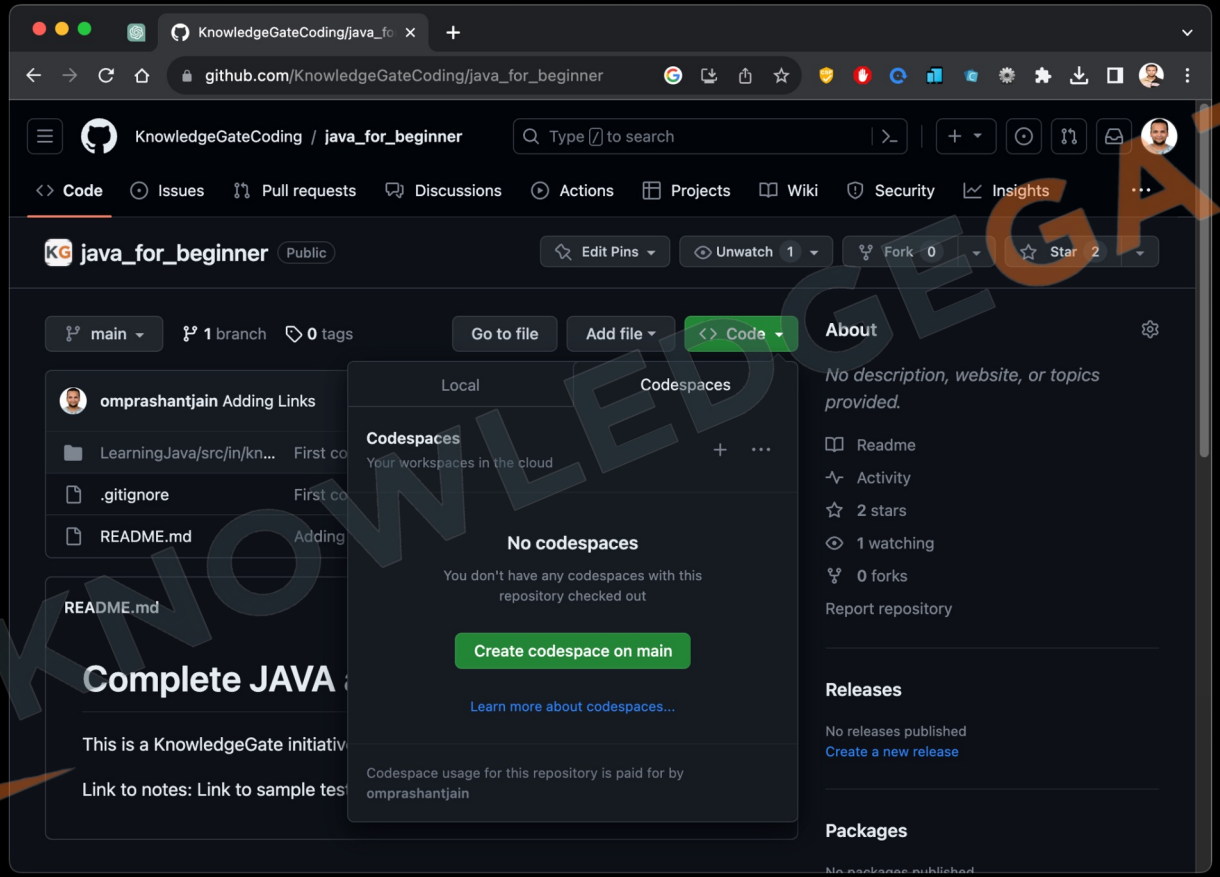
1.3 Account Creation



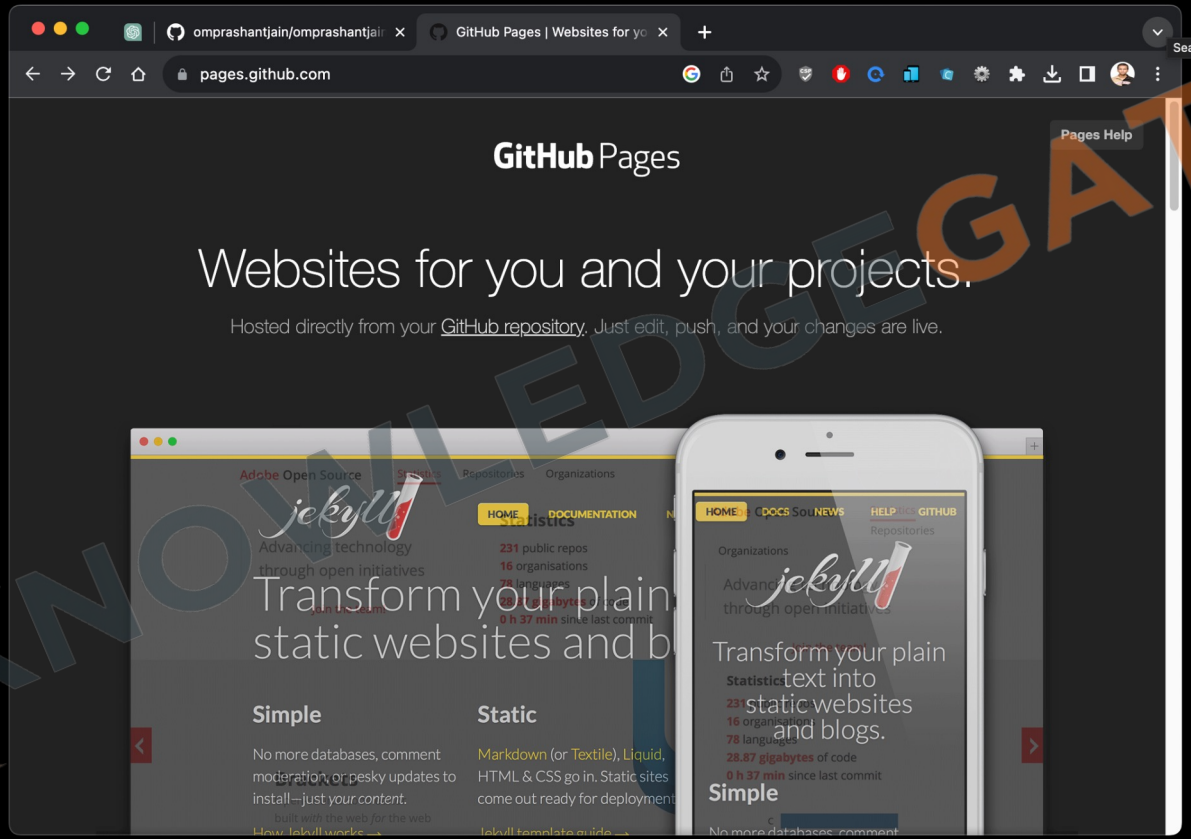
1.4 Creating a Repo



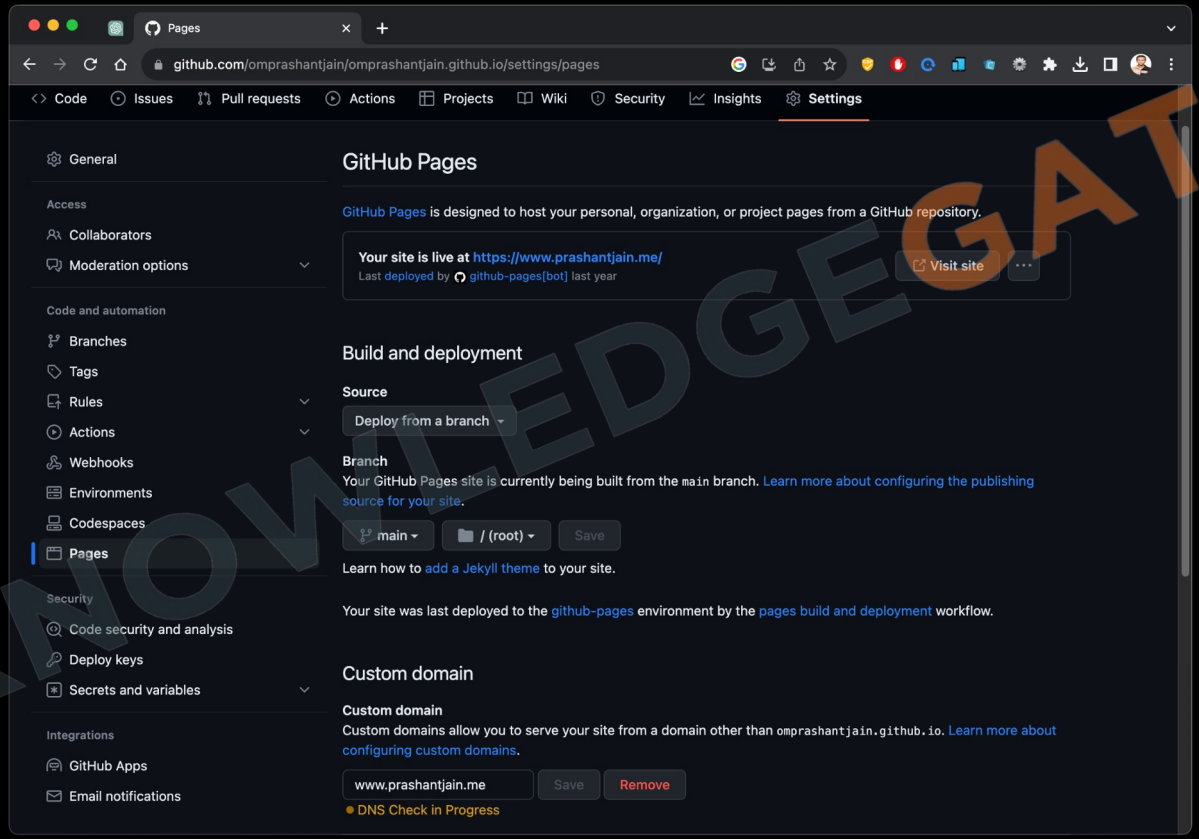
1.5 Creating a CodeSpace



1.6 Creating a Github Page



1.7 Publishing our Project



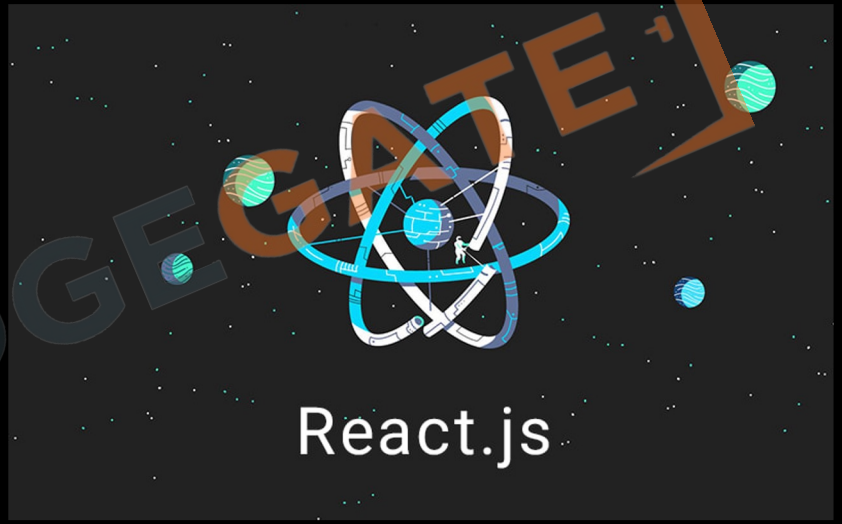
Level Bonus

Github Pages & CodeSpace

2. Frameworks

3.1 ReactJS

1. Definition: A tool for making websites interactive.
2. Components: Reusable pieces for building a webpage.
3. Virtual DOM: Makes websites faster by updating only what's needed.
4. JSX: A special way to write code that looks like HTML.
5. State: Keeps track of changes on the webpage.
6. Props: Shares information between different parts of a webpage.



3.2 AngularJS

1. Definition: A framework for building web applications, developed by Google.
2. Two-Way Data Binding: Updates both the view and the model simultaneously.
3. Directives: Custom HTML tags for added functionality.
4. Dependency Injection: Automatically manages how parts of the app work together.
5. Controllers: Manages the data for a specific part of the webpage.
6. SPA Support: Good for Single Page Applications where the page doesn't reload.

3.3 VueJS

1. Definition: A JavaScript framework for creating web interfaces.
2. Components: Small, reusable parts for building a website.
3. Reactivity: Automatically updates the webpage when data changes.
4. Directives: Special tokens in HTML for added functionality.
5. Vuex: Helps manage shared data across the site.
6. Single-File Components: Keeps template, script, and style in one file.

Level Bonus Revision

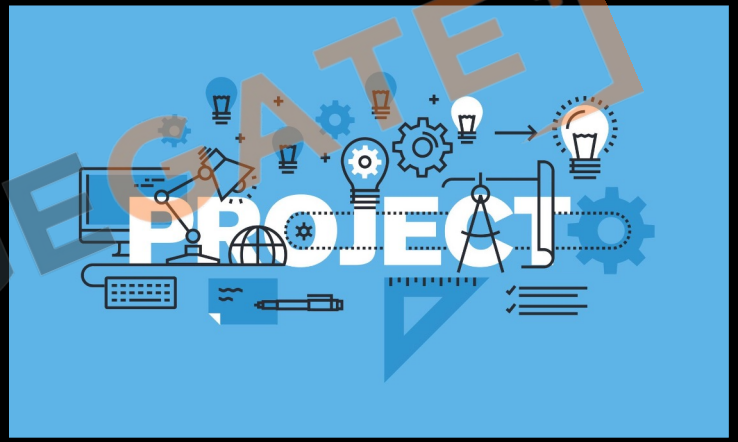
Github Pages & CodeSpace

1. Github
    1. What is Version Control
    2. What is Git and GitHub
    3. Account Creation
    4. Creating a Repo
    5. Creating a Codespace
    6. Creating a Github page
    7. Publishing our project
2. FrameWorks
    1. React
    2. Angular
    3. Vue

Level Bonus Project

Github Pages & CodeSpace

1. Create a Github account if you don’t have one.
2. Create a repo for the project you have done till now.
3. Create a Codespace and make changes.
4. Publish it using Github pages.



Major Project

Idea

1. Create your Resume or Portfolio website.
2. Create a repo in Github.
3. Publish it on Github pages.
4. Add the link to your resume.
5. Add the link in the comment section.