

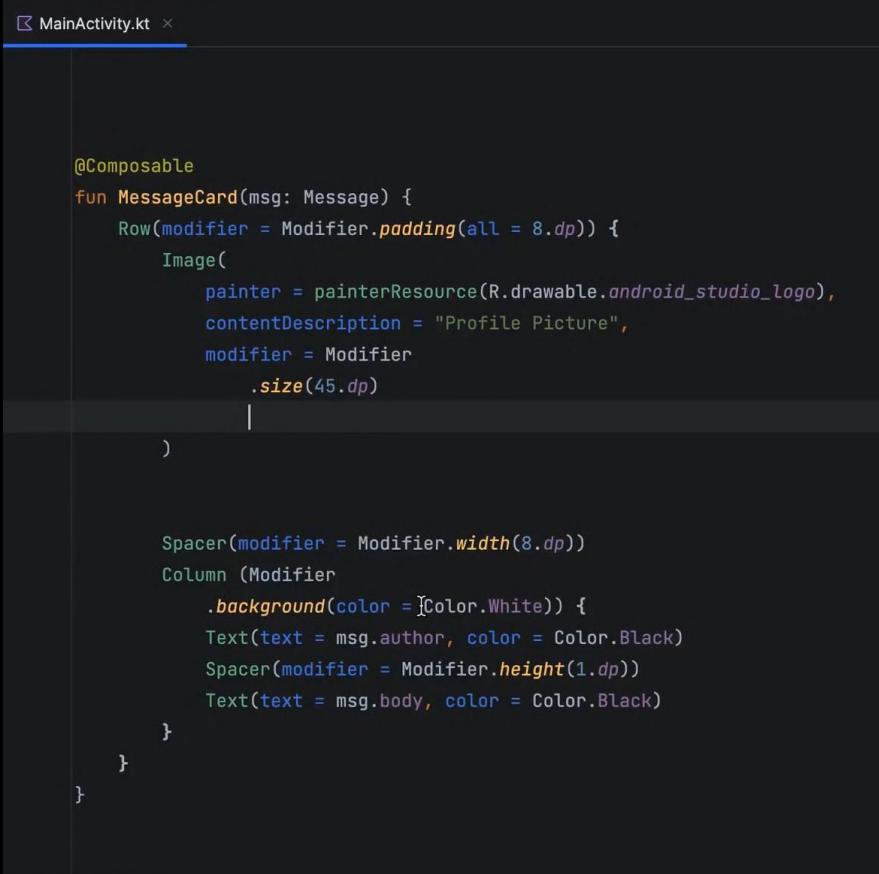
1.1 What is IDE

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



The screenshot shows a dark-themed code editor window titled "MainActivity.kt". The code is a Composable function named "MessageCard". It uses the Material Components library's Row and Column components to layout a profile picture and text. The code includes syntax highlighting for keywords like @Composable, fun, Row, Image, painter, contentDescription, modifier, size, Spacer, Column, background, Text, and height. Braces and line numbers are also visible.

```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.android_studio_logo),
            contentDescription = "Profile Picture",
            modifier = Modifier
                .size(45.dp)
        )
        Spacer(modifier = Modifier.width(8.dp))
        Column (Modifier
            .background(color = Color.White)) {
            Text(text = msg.author, color = Color.Black)
            Spacer(modifier = Modifier.height(1.dp))
            Text(text = msg.body, color = Color.Black)
        }
    }
}
```



1.3 IDE Selection

1. Sublime Text

2. Atom

3. VS Code

4. Github CodeSpaces



The screenshot shows the Visual Studio Code interface. On the left, the Extensions Marketplace is open, displaying various extensions like Python, GitLens, C/C++, ESLint, Debugger for C#, Language Support, vscode-icons, and Vetur. The main code editor window on the right shows a file named 'blog-post.js' with some JavaScript code. Below the editor is the Terminal panel, which shows the command 'npm run develop' being run and completed successfully. The status bar at the bottom indicates 'Spaces: 2'.

```
src > components > JS blog-post.js > ↗<function> > ↗blogPost
1 import { graphql } from "gatsby"
2 import React from "react"
3 import Image from "gatsby-image"
4
5 export default ({ data }) => {
6   const blogPost = data.cms.blogPost
[el] data
[el] dateFormat
[el] debug
[el] debugger
[el] decodeURI
[el] decodeURIComponent
[el] default
[el] defaultCenter
[el] delete
[el] departFocus
[el] devicePixelRatio
[el] dispatchEvent
19 }
20
21 export const query = graphql`var data: any
info i [wdm]: Compiling...
DONE Compiled successfully in 70ms
4:05:11 PM
info i [wdm]:
info i [wdm]: Compiled successfully.
Spaces: 2  UTF-8  LF  JavaScript  ⚡  🔍
```

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



Full Stack

2.3 Role of Browser



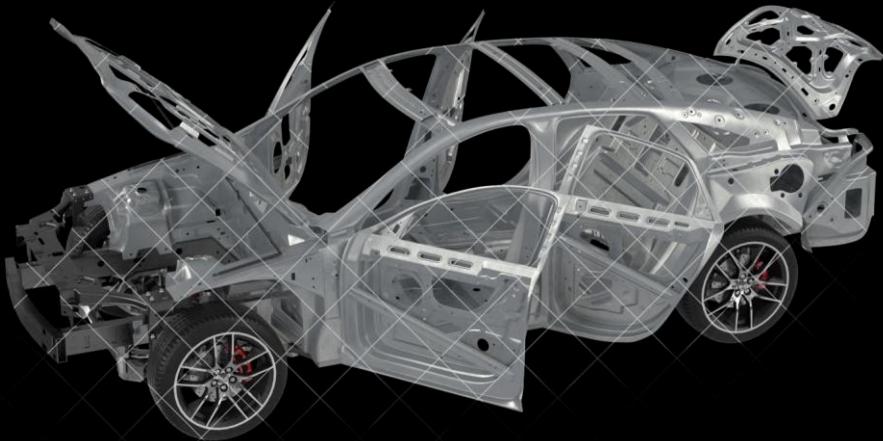
1. **Displays Web Page:** Turns HTML code into what you see on screen.
2. **User Clicks:** Helps you interact with the web page.
3. **Updates Content:** Allows changes to the page using JavaScript.
4. **Loads Files:** Gets HTML, images, etc., from the server.



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 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

1.2 File Extension

HTML

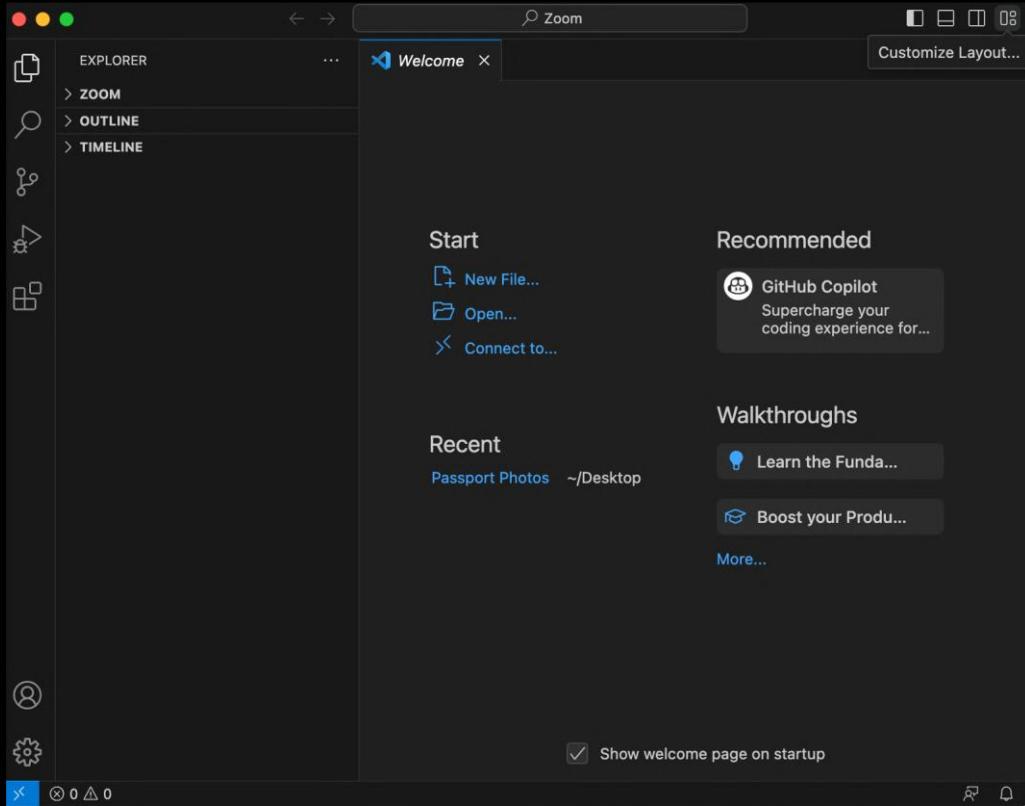
1. Most commonly used.
2. Works across all browsers.
3. Widely recognized and supported.
4. Typically saved as .html.



HTM

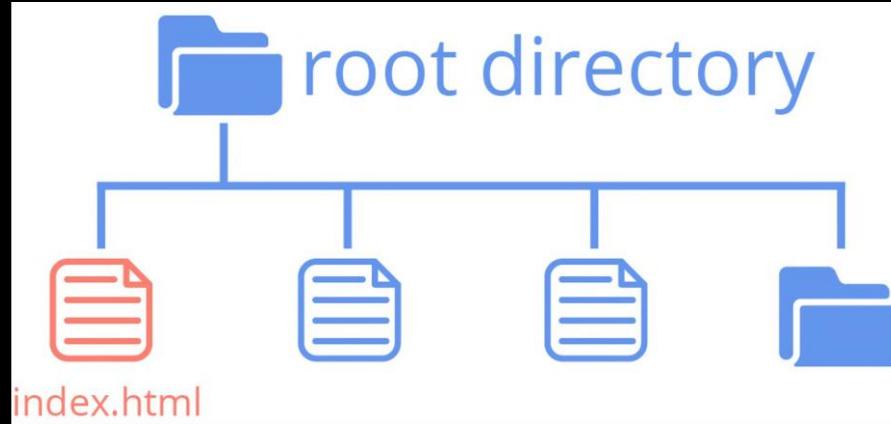
1. Less commonly used.
2. Originated for compatibility with older systems.
3. Works same as .html.
4. Typically saved as .htm.

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

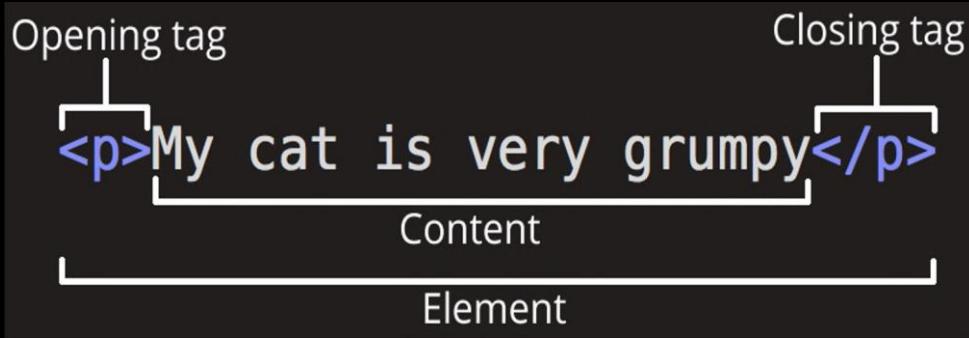
HTML Basics



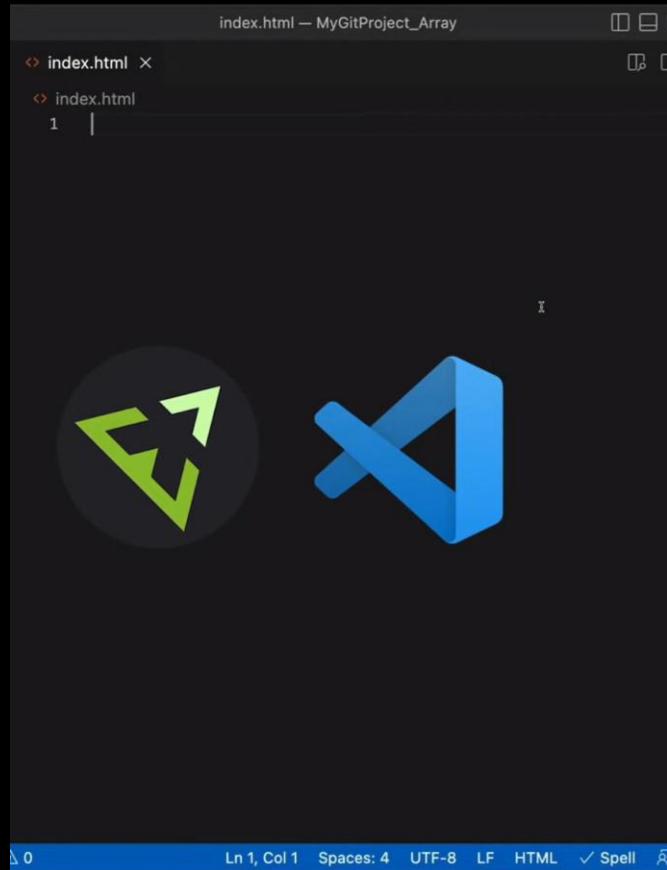
2. Basics of HTML

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.

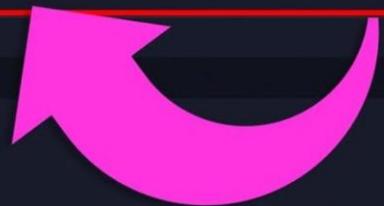
2.3 Basic HTML Page

```
<!DOCTYPE html>           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>
```

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>
```



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**

Writing comments in HTML

Single-line Comment

```
1 <!--This is a single line  
comment in HTML. You cannot  
see it on a webpage. Click  
on view-source to see a  
message I left just for you.  
-->
```

```
1 <!-- This is a multi-line  
comment in HTML.  
2 You cannot see it on a  
webpage.  
3 If you view-source on the  
browser you can see the  
comment there.-->
```

Multi-line Comment

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>`

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

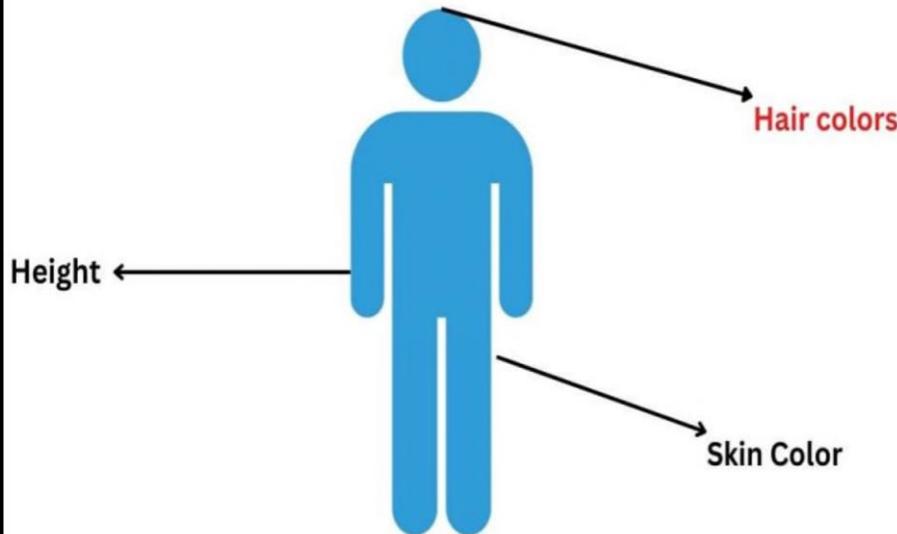
3. Character Entity Reference

1. What are Character Entity References

Level 2

Must-Use HTML Tags

Attribute



1. HTML Attributes

1.1 What are HTML Attributes?

Html Attributes

Attribute
`<tag attribute="value">Text Content </tag>`

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

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

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

The browser window shows the rendered HTML with the text "This is the header" displayed.

- **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.

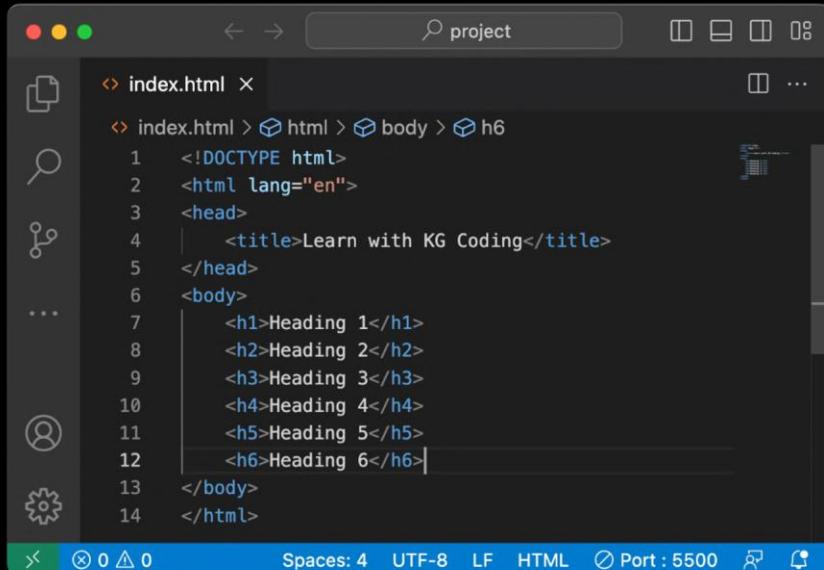
Level 2

Must-Use HTML Tags



2. HTML Tags

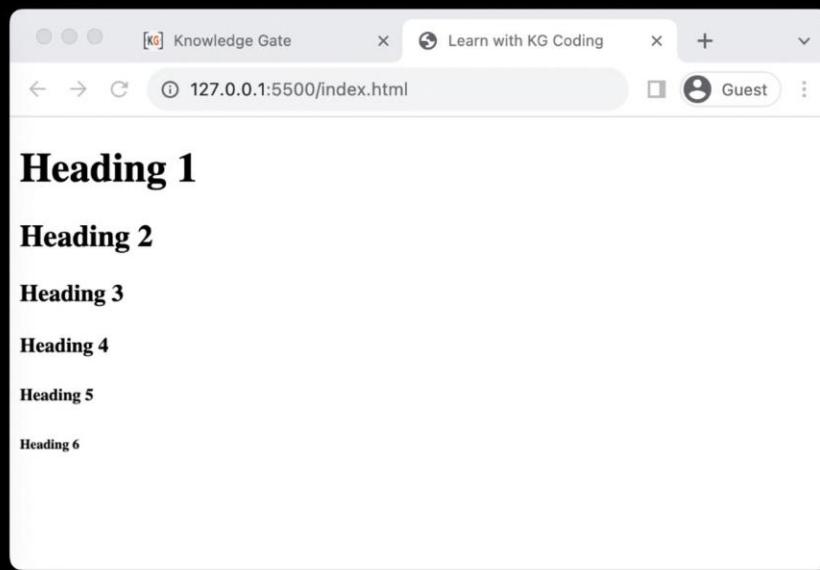
2.1 Heading Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, displaying the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Learn with KG Coding</title>
5   </head>
6   <body>
7     <h1>Heading 1</h1>
8     <h2>Heading 2</h2>
9     <h3>Heading 3</h3>
10    <h4>Heading 4</h4>
11    <h5>Heading 5</h5>
12    <h6>Heading 6</h6>
13  </body>
14 </html>
```

The code editor interface includes a sidebar with icons for file, search, and help, and a bottom bar with settings like "Spaces: 4", "UTF-8", "LF", "HTML", and "Port : 5500".



A screenshot of a web browser window titled "Knowledge Gate" showing the rendered content of "index.html". The page displays the following headings:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

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

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:

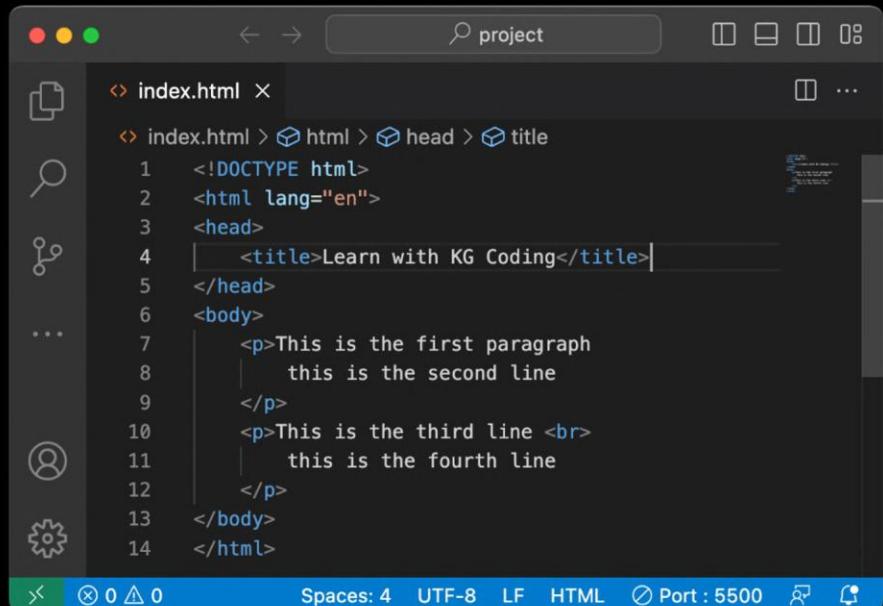
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <p>This is the first paragraph</p>
    <p>This is the Second paragraph</p>
    <p>This is the Third paragraph</p>
</body>
</html>
```

The browser window shows the rendered HTML with three paragraphs:

This is the first paragraph
This is the Second paragraph
This is the Third paragraph

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

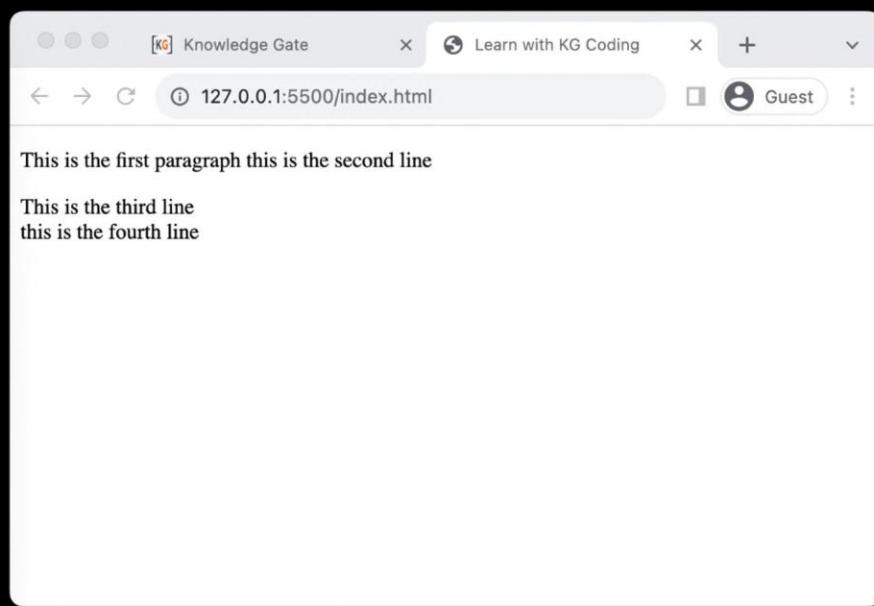
2.3
 Tag



A screenshot of a code editor window titled "index.html". The code is as follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <p>This is the first paragraph
        this is the second line
    </p>
    <p>This is the third line <br>
        this is the fourth line
    </p>
</body>
</html>
```

The code editor has a dark theme with light-colored syntax highlighting. It includes a sidebar with various icons for file operations like copy, paste, search, and refresh.

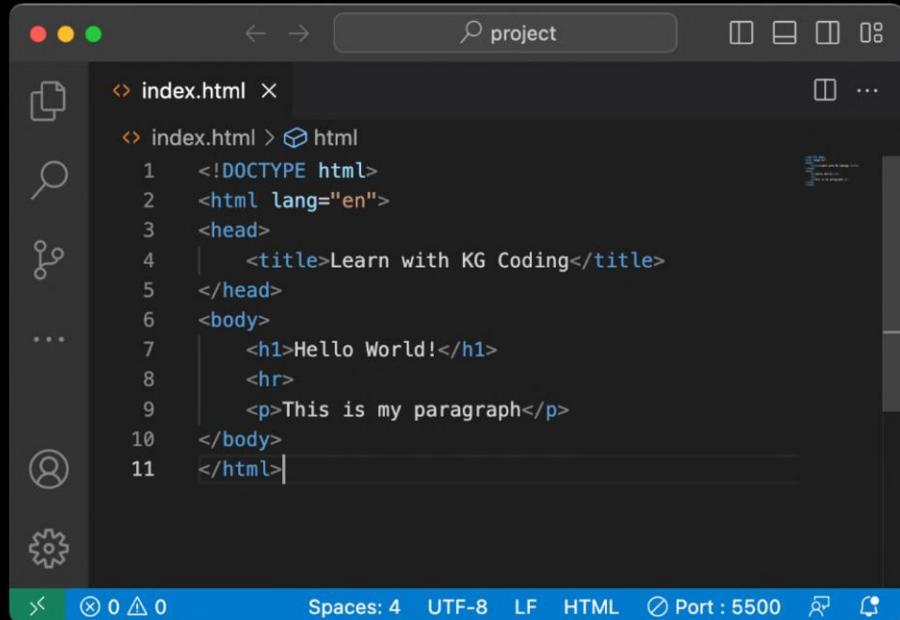


A screenshot of a web browser window titled "Knowledge Gate". The URL is "127.0.0.1:5500/index.html". The page content is:

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

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

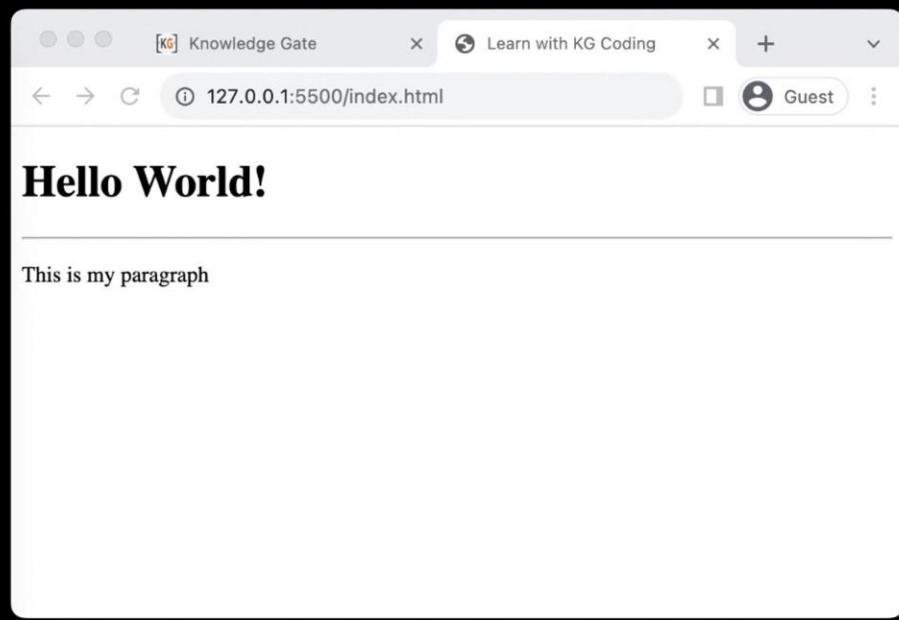
2.3 <HR> Tag



A screenshot of a code editor window titled "index.html". The code is as follows:

```
<!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 editor has a dark theme with light-colored syntax highlighting. It includes a sidebar with various icons and a bottom status bar showing "Spaces: 4" and "Port : 5500".



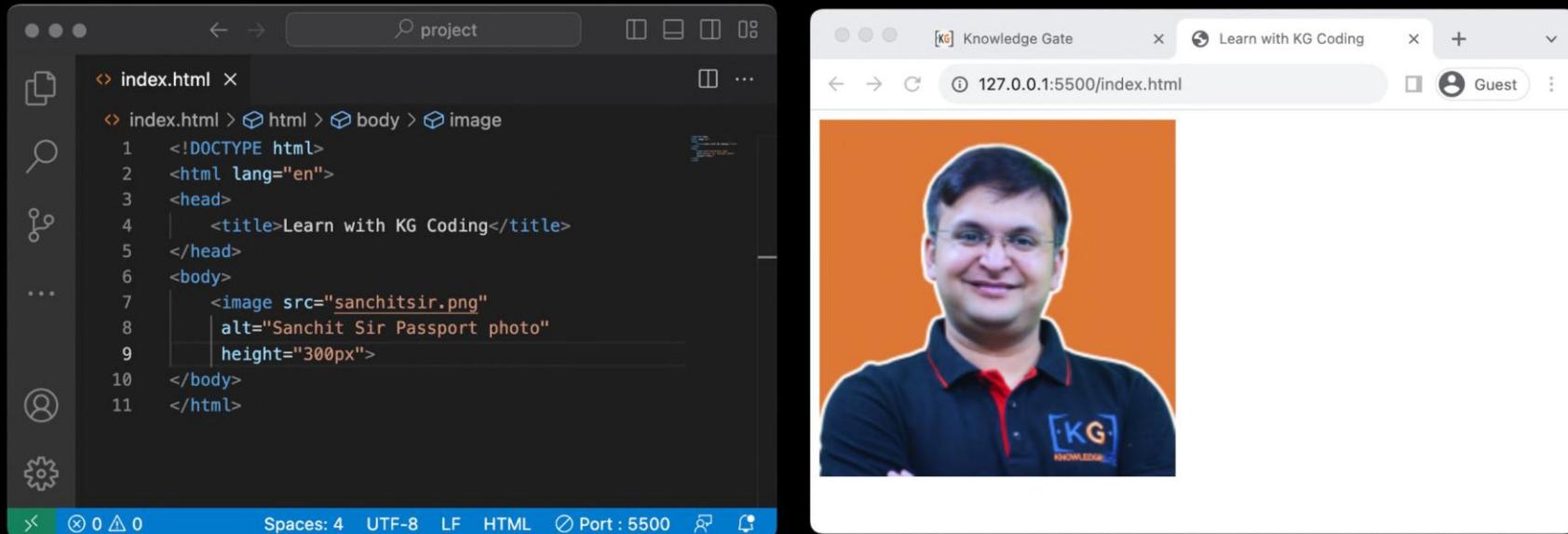
A screenshot of a web browser window titled "Knowledge Gate" showing the URL "127.0.0.1:5500/index.html". The page displays the following content:

Hello World!

This is my paragraph

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

2.4 Image Tag



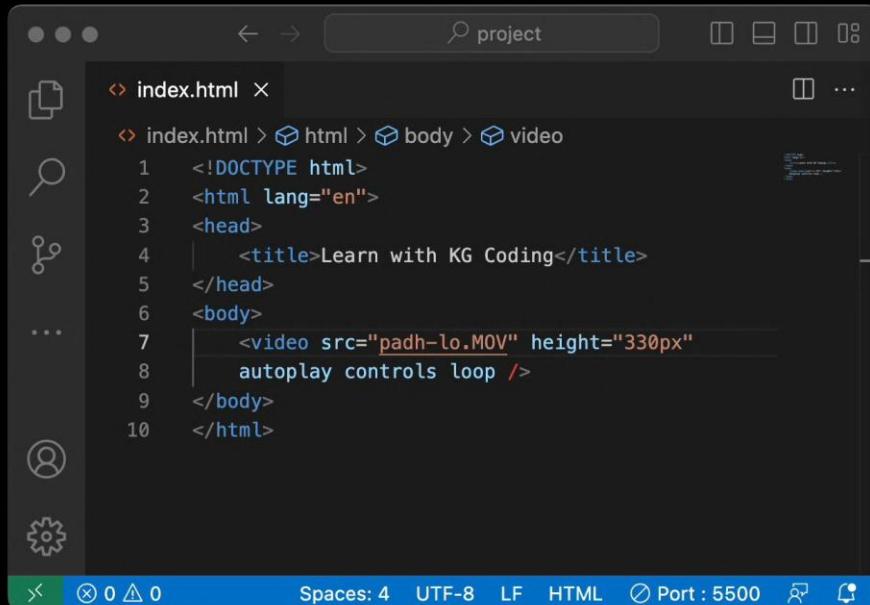
The image shows a code editor on the left and a web browser on the right. The code editor displays the contents of index.html:

```
<!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>
```

The browser window shows the rendered output of the code, displaying a portrait of a man with glasses and a blue polo shirt featuring a 'KG' logo, set against an orange background.

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

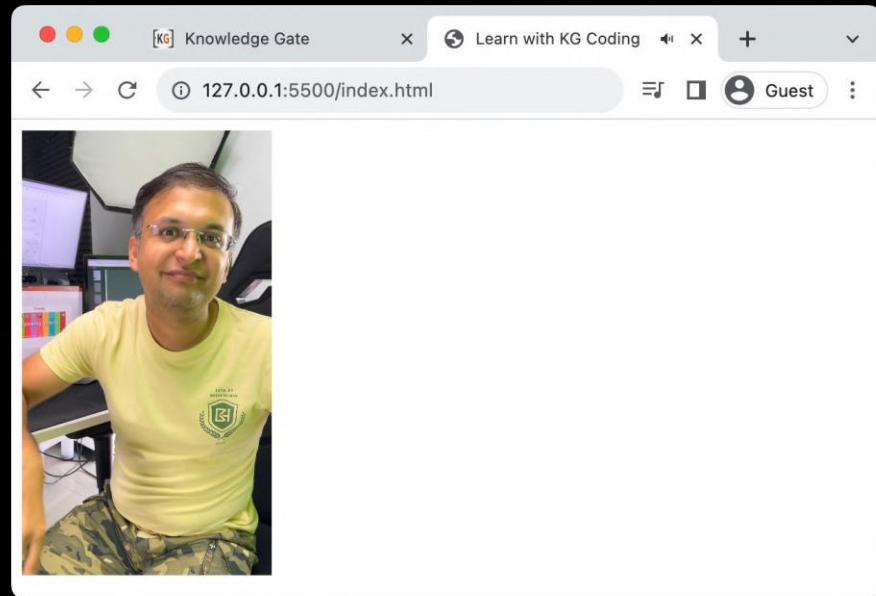
2.5 Video Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following code:

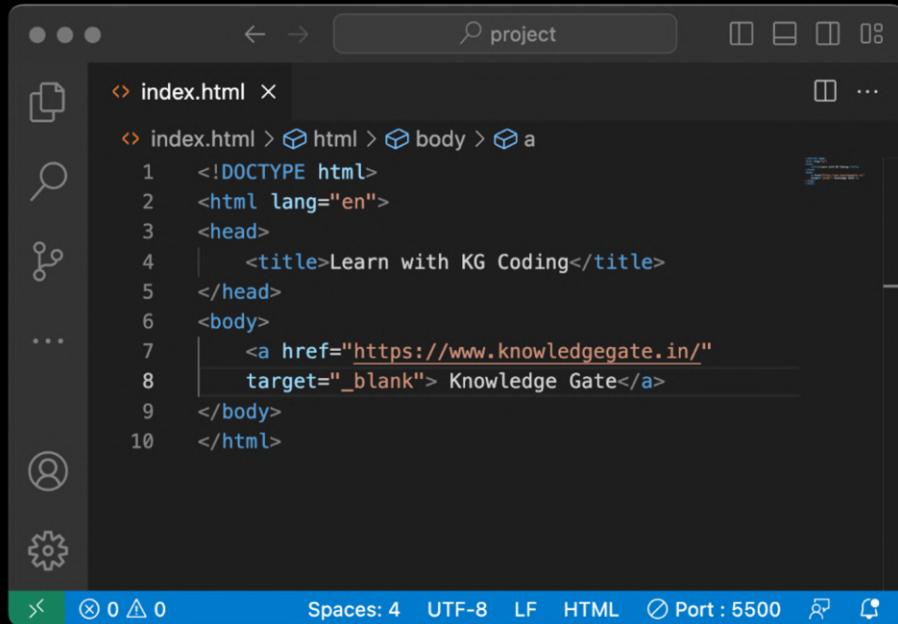
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <video src="padh-lo.MOV" height="330px"
        autoplay controls loop />
</body>
</html>
```

The code editor has a dark theme with light-colored syntax highlighting. The status bar at the bottom shows "Spaces: 4" and "Port : 5500".



1. Embeds video files on a page
2. Uses **src** attribute for video URL
3. Supports **multiple formats** like MP4, WebM
4. Allows for built-in controls via attributes like **autoplay, controls, loop**

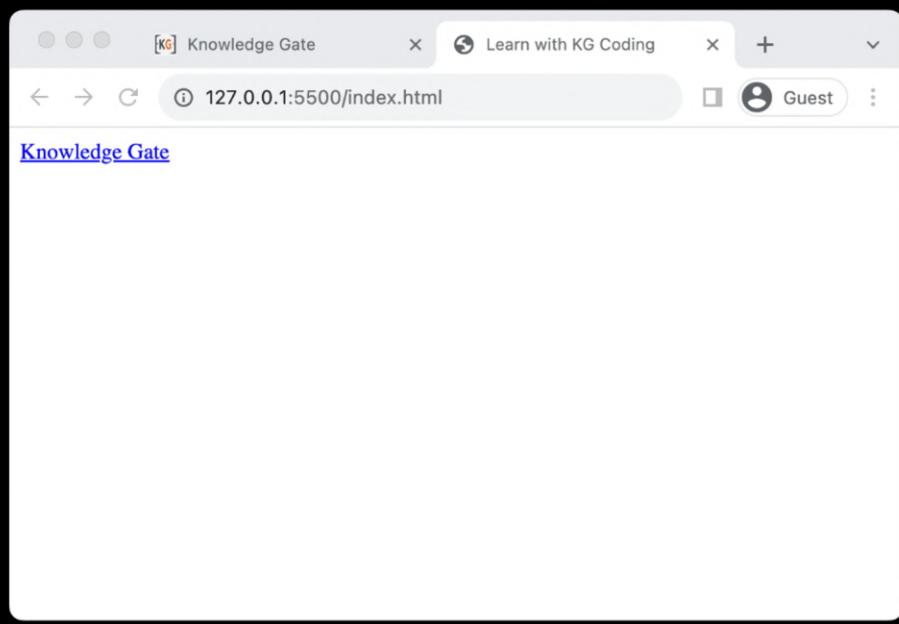
2.6 Anchor Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following HTML code:

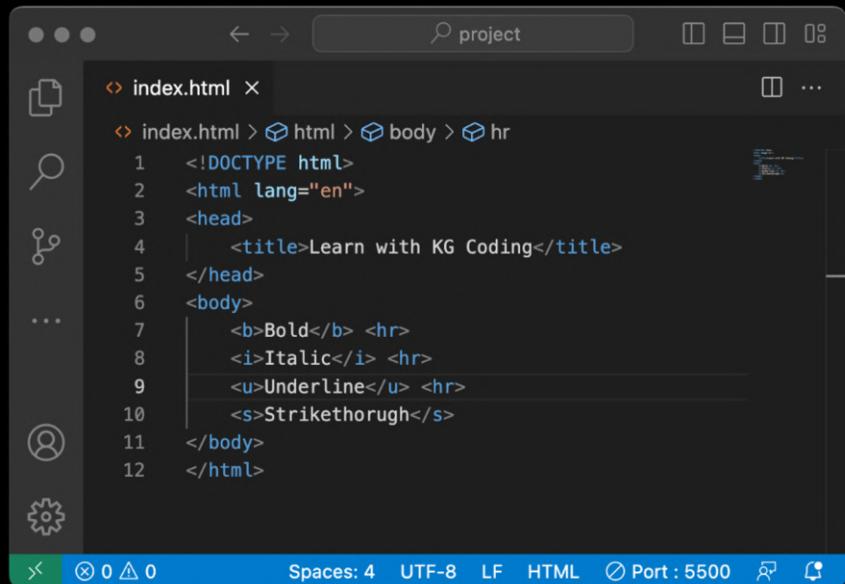
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <a href="https://www.knowledgegate.in/" target="_blank"> Knowledge Gate</a>
</body>
</html>
```

The code editor has a dark theme with light-colored syntax highlighting. The sidebar on the left contains icons for file operations like new, open, save, and search.



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

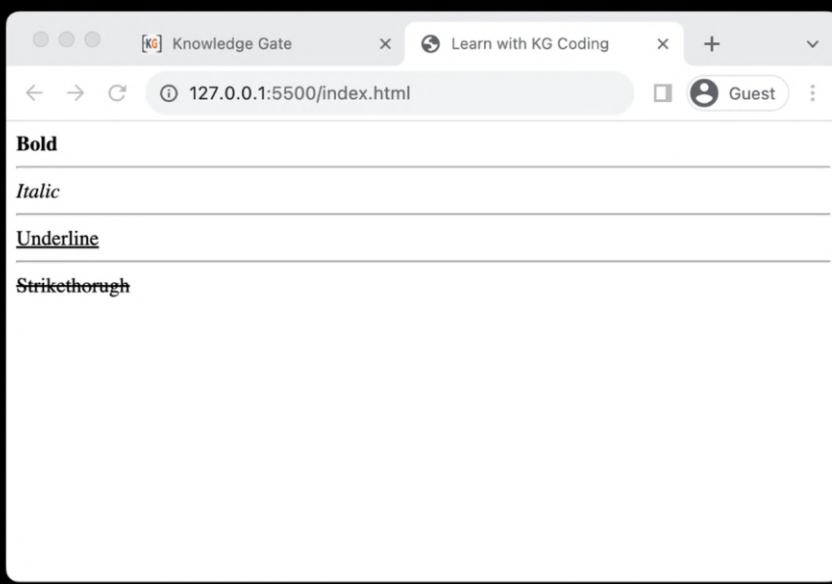
2.7 Bold/Italic/Underline/Strikethrough Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, displaying the following HTML code:

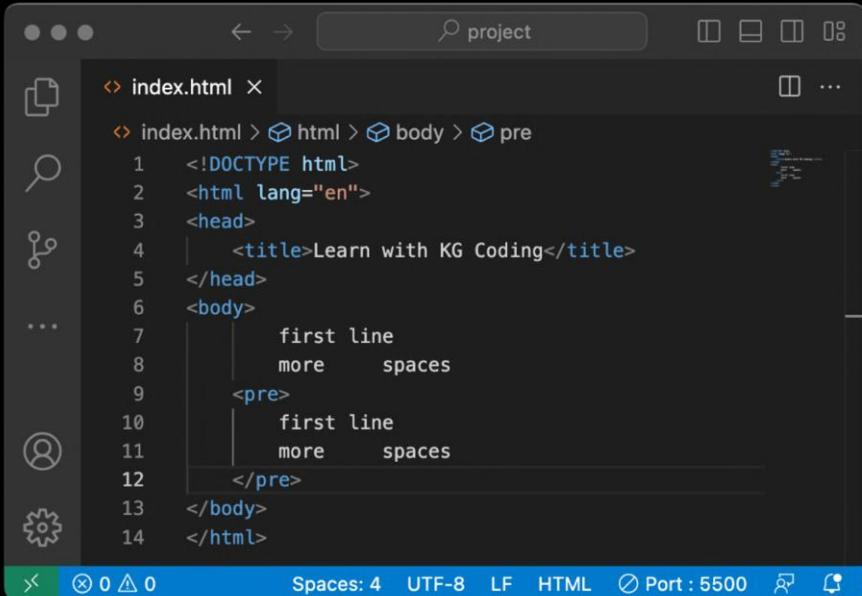
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <b>Bold</b> <hr>
    <i>Italic</i> <hr>
    <u>Underline</u> <hr>
    <s>Strikethrough</s>
</body>
</html>
```

The code editor has a dark theme with light-colored syntax highlighting. It includes a sidebar with various icons and a bottom status bar showing "Spaces: 4", "UTF-8", "LF", "HTML", and "Port : 5500".



1. **** 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

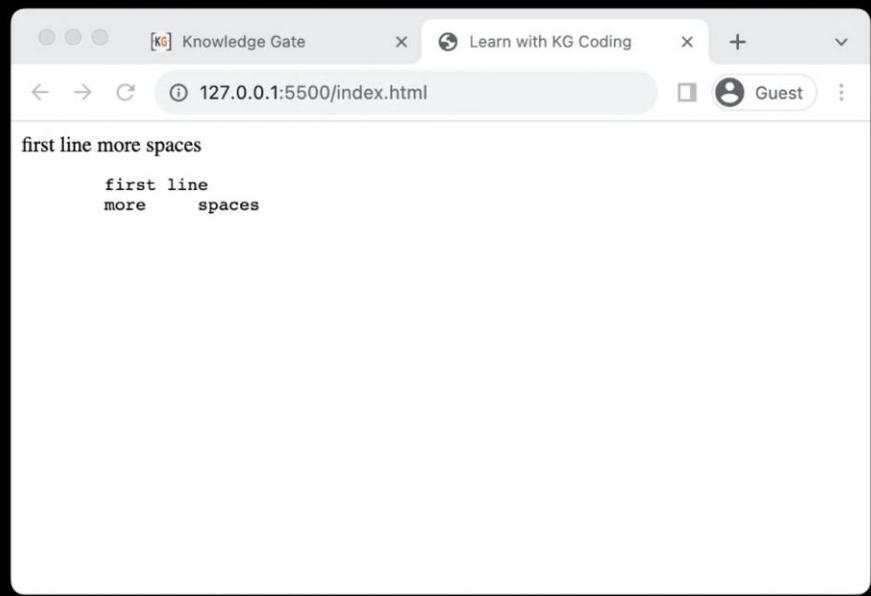
2.8 Pre Tag



A screenshot of a code editor window titled "index.html X". The code editor shows the following HTML structure:

```
<!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 code editor has a dark theme with light-colored text. It includes a sidebar with icons for file operations, search, and user profile. The bottom bar shows file statistics (0 lines, 0 words, 0 characters) and encoding information (Spaces: 4, UTF-8, LF, HTML). A port number (Port : 5500) is also visible.



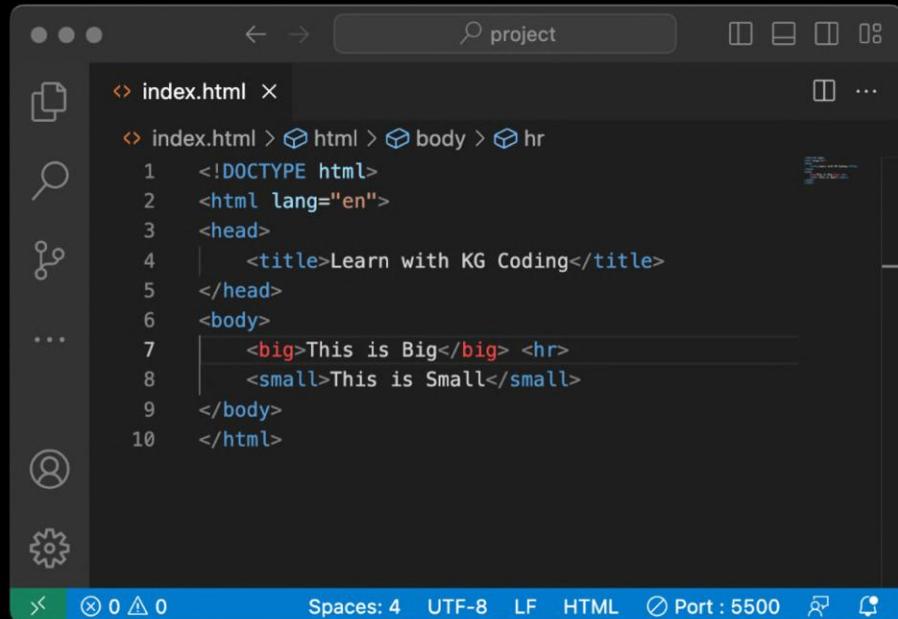
A screenshot of a web browser window titled "Knowledge Gate" showing the URL "127.0.0.1:5500/index.html". The browser displays the following text:

first line more spaces
first line
more spaces

The browser interface includes standard navigation buttons, a search bar, and a user profile icon.

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

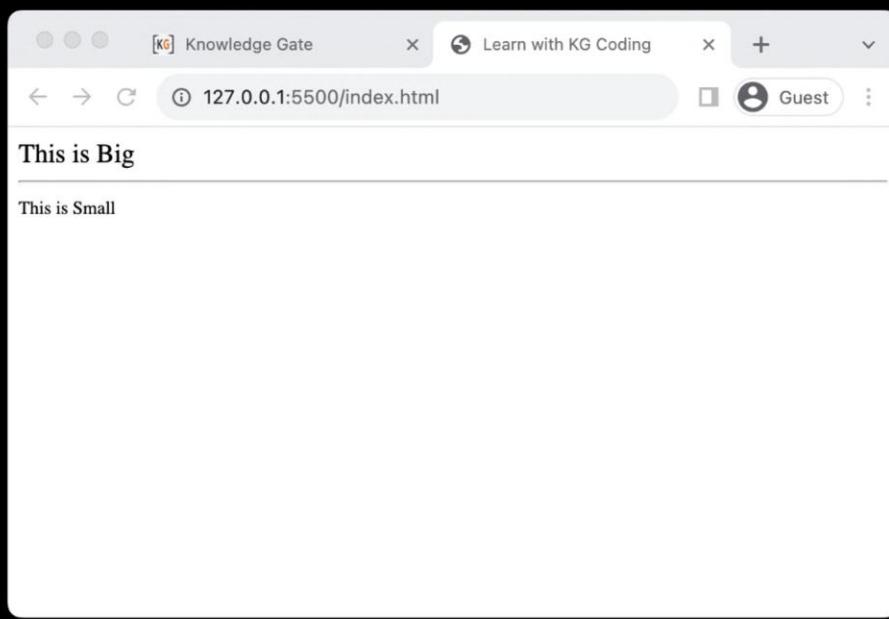
2.9 Big/Small Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following code:

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

The code editor has a dark theme with light-colored syntax highlighting. The sidebar on the left contains icons for file operations like new, open, save, and search.



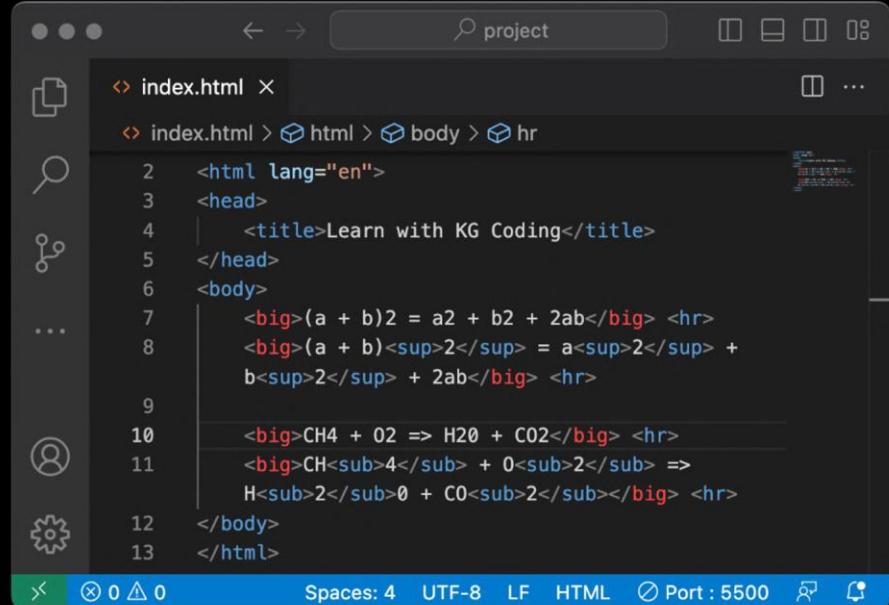
A screenshot of a web browser window titled "Knowledge Gate". The address bar shows "127.0.0.1:5500/index.html". The page content is:

This is Big

This is Small

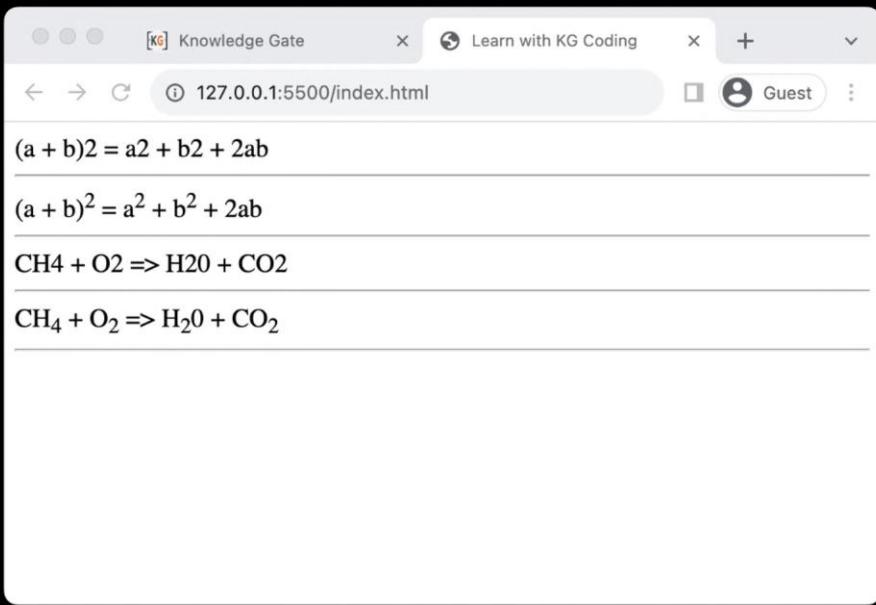
1. **<big>** increases text size
2. **<small>** decreases text size
3. Less common due to CSS alternatives

2.10 Superscript/Subscript Tag



A screenshot of a code editor window titled "index.html". The code is as follows:

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



A screenshot of a web browser window titled "Knowledge Gate". The URL is "127.0.0.1:5500/index.html". The page displays the following text:

(a + b)² = a² + b² + 2ab
(a + b)² = a² + b² + 2ab
CH₄ + O₂ => H₂O + CO₂
CH₄ + O₂ => H₂O + CO₂

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 &)

 	–	–	—	−	°	°	Δ	Δ	α	&al;	
€	€	—	—	±	±	°	º	Λ	Λ	β	&be;
ƒ	¢	…	…	√	√	ª	ª	Θ	Θ	γ	&ga;
£	£	§	§	∞	∞	¹	¹	Ξ	Ξ	δ	&de;
¥	¥	¶	¶	∞	∝	²	²	Π	Π	ε	&ep;
¤	¤	†	†	×	×	³	³	Σ	Σ	ζ	ζ
f	ƒ	‡	‡	÷	÷	¼	¼	Φ	Φ	η	&et;
©	©	®	¡	~	∼	½	½	Ψ	Ψ	θ	&th;
®	®	®	¿	≈	≈	¾	¾	Ω	Ω	ι	&iot;
™	™	%	‰	≡	≅	∴	∴	∇	∇	κ	κ

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

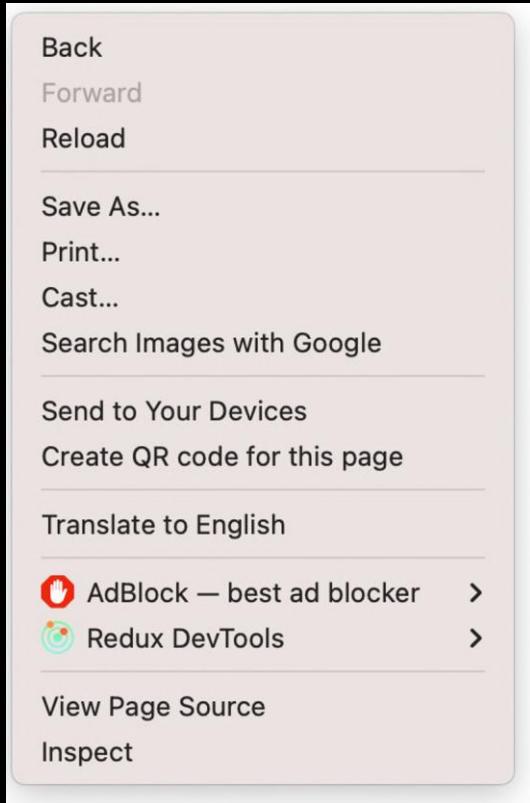
Level 3

Browser Tools



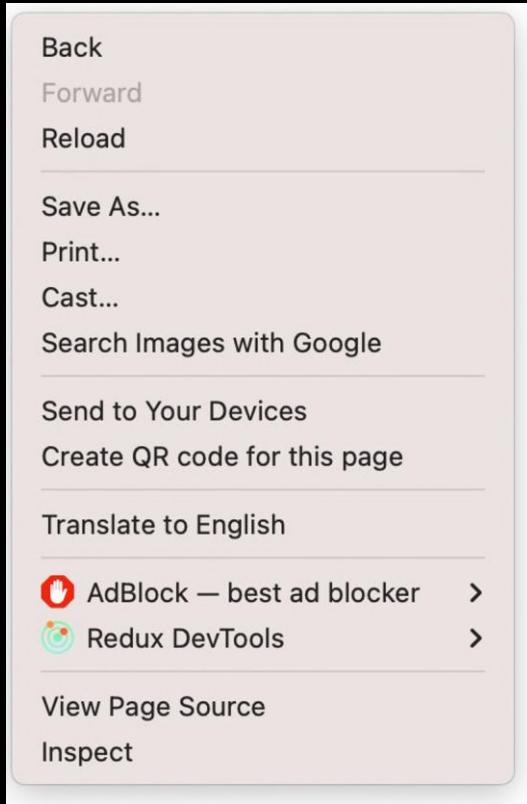
1. Browser Tools

1.1 View Page Source



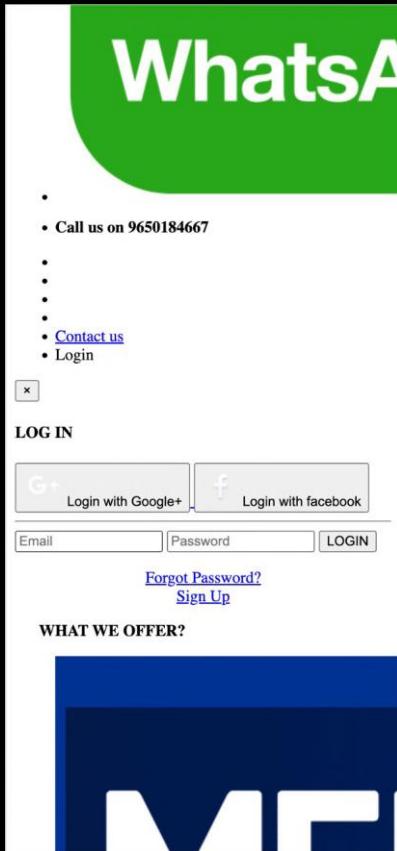
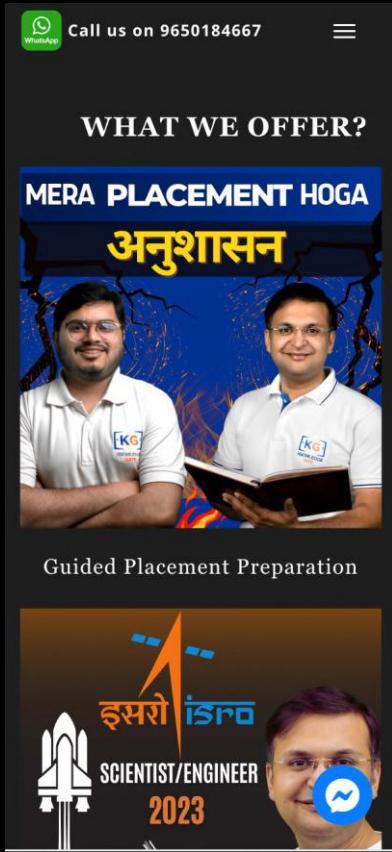
1. Displays raw HTML and CSS
2. Useful for debugging and learning
3. Shows external files like JavaScript links

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

1.3 HTML without CSS



Level 3

Browser Tools



2. Responsive Design

2.1 Different Screen Sizes

Responsive

iPhone SE
iPhone XR
✓ iPhone 12 Pro
Pixel 5
Samsung Galaxy S8+
Samsung Galaxy S20 Ultra
iPad Air
iPad Mini
Surface Pro 7
Surface Duo
Galaxy Fold
Samsung Galaxy A51/71
Nest Hub
Nest Hub Max
Facebook for Android v407 on Pixel 6

Edit...

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

3.1 Live edit HTML

The screenshot shows the YouTube channel profile for 'KG Coding'. The channel banner features a man in a white t-shirt with 'KNOLLEGATE' printed on it, surrounded by various technology logos like Angular, Vue.js, MERN, C++, C#, and Java. The channel name 'KG Coding' is displayed in large letters, with the handle '@KG_Coding' below it. It shows 10M subscribers and 1 video. A 'Coming Soon..... >' message is present. Below the banner, there are tabs for HOME, VIDEOS, LIVE, and PLAYLISTS. A section titled 'Our Other Channels' lists 'KNOWLEDGE GATE' (594K subscribers) and 'KnowledgeGate Placement Prep' (105K subscribers). The right side of the screen displays the full URL of the page in the browser's address bar: 'https://www.youtube.com/@KG_Coding?utm_source=yt_mainpage&utm_medium=channel_header&utm_campaign=channel_header'.

Changed **Subscriber** count

3.2 Live edit CSS

The screenshot shows a YouTube channel page for 'kg coding'. The channel has 10M subscribers and 1 video. The bio reads 'Coming Soon..... >' and includes links to 'knowledgegate.in' and '1 more link'. Below the bio, there are four navigation tabs: HOME (selected), VIDEOS, LIVE, and PLAYLISTS. At the bottom, there are two recommended channels: 'KNOWLEDGE GATE' (594K subscribers) and 'KnowledgeGate Placement Prep' (105K subscribers). A sidebar on the right displays the raw HTML code of the channel's header area.

```
<div>
  <div>
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        <div>
                          <div>
                            <div>
                              <div>
                                <div>
                                  <div>
                                    <div>
                                      <div>
                                        <div>
                                          <div>
                                            <div>
                                              <div>
                                                <div>
                                                  <div>
                                                    <div>
                                                      <div>
                                                        <div>
                                                          <div>
                                                            <div>
                                                              <div>
                                                                <div>
                                                                  <div>
                                                                    <div>
                                                                      <div>
                                                                        <div>
                                                                          <div>
                                                                            <div>
                                                                              <div>
                                                                                <div>
                                                                                  <div>
                                                                                    <div>
                                                                                      <div>
                                                                                        <div>
                                                                                          <div>
                                                                                            <div>
                                                                                              <div>
                                                                                                <div>
                                                                                                  <div>
                                                                                                    <div>
                                                                                                      <div>
                                                                                                        <div>
                                                                                                          <div>
                                                                                                            <div>
                                                                                                              <div>
                                                                                                                <div>
                                                                                                                  <div>
                                                                                                                    <div>
                                                                                                                      <div>
                                                                                                                        <div>
                                                                                                                          <div>
                                                                                                                            <div>
                                                                                                                              <div>
                                                                                                                                <div>
                                                                                                                                  <div>
                                                                                                                                    <div>
                                                                                                                                      <div>
                                                                                                                                        <div>
                                                                                                                                          <div>
                                                                                                                                            <div>
                                                                                                                                              <div>
                                                                                                                                                <div>
                                                                                                                                                  <div>
                                                                                                                                                    <div>
                                                                                                      <div>
                                                                                                        <div>
                                                                                                          <div>
                                                                                                            <div>
                                                                                                              <div>
                                                                                                                <div>
                                                                                                                  <div>
                                                                                                                    <div>
                                                                                                                      <div>
                                                                                                                        <div>
                                                                                                                          <div>
                                                                                                                            <div>
                                                                                                                              <div>
                                                                                                                                <div>
                                                                                                                                  <div>
                                                                                                                                    <div>
                                                                                                                                      <div>
                                                                                                                                        <div>
                                                                                                                                          <div>
                                                                                                                                            <div>
................................................................
```

Changed Channel
Name color

3.3 Live edit JS



The screenshot shows a YouTube video player interface. On the left, there's a sidebar with navigation links: Home, Shorts, Subscriptions, and Library. Below these are four circular profile pictures. At the bottom, there are four tabs: HOME, VIDEOS, LIVE, and PLAYLISTS.

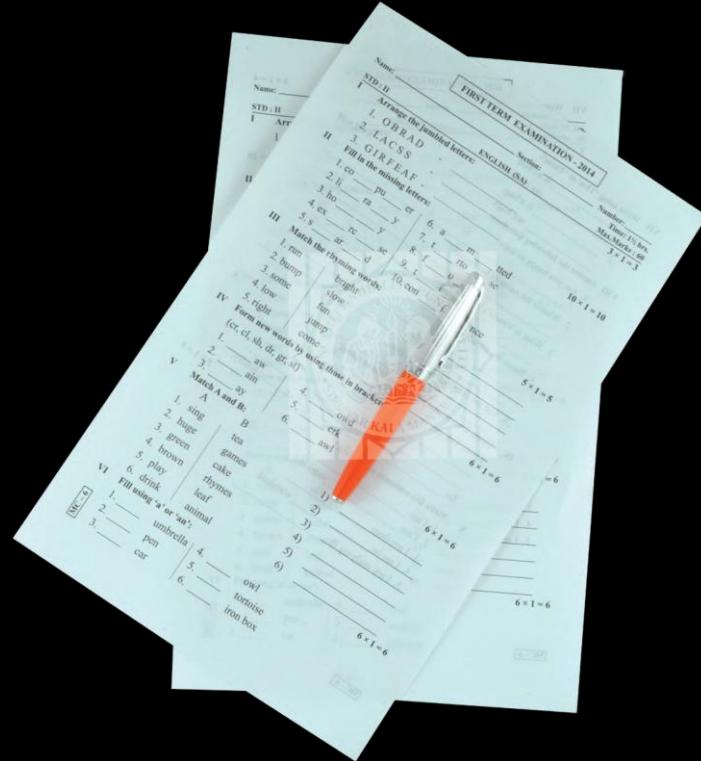
The main area displays a video thumbnail featuring a developer with glasses and a white t-shirt, standing in front of a banner that says "CODING MAD". The banner includes logos for Angular, Vue.js, MongoDB, Express, React, Node.js, and C/C++. A search bar is at the top of the video player.

To the right of the video player is a browser developer tools window. The "Console" tab is selected, showing a list of network requests and their status. Some entries show errors like "ERR_BLOCKED_BY_CLIENT" and "ERR_FAILED". Other entries show successful requests like "GET https://static.doubleclick.net/instream/ad_status.js". The console also displays some JavaScript code snippets and a warning about a preloaded resource.

```
com/s/desktop/462a8d5d/jsbin/custom-elements-es5-adapter.js:1:108
vascascript/custom_elements/fast-shim.js.sourcemap: HTTP error: status code 404,
net::ERR_HTTP_RESPONSE_CODE_FAILURE
⚠ DevTools failed to load source map: Could not load content for https://www.youtube.
com/s/desktop/462a8d5d/jsbin/web-animations-next-lite.min.vflset/web-animations-nex
t-lite.min.js.map: HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
⚠ DevTools failed to load source map: Could not load content for https://www.youtube.
com/s/desktop/462a8d5d/jsbin/webcomponents-sd.vflset/bl..._party/javascript/polymer/v
2/webcomponentsjs/webcomponents-sd.js.sourcemap: HTTP error: status code 404,
net::ERR_HTTP_RESPONSE_CODE_FAILURE
LegacyDataMixin will be applied to all desktop_polymer_enable_wl_icons.js:4482
legacy elements.
Set '_legacyUndefinedCheck: true` on element class to enable.
x GET https://googleads.g.doubleclick.net/pagead/id
net::ERR_BLOCKED_BY_CLIENT
VM15538:201
x GET https://www.google.com/pagead/lv www.google.com/pagea...5CoWhrV 7RpGsmwVg:1
z?evtid=ACd6KtzZ40h0LZgbnYYZ8gxW6cxP
BX2-4..inAppBootstrap%3AUnclassified&aaz=1&sigh=AB9vU40PfuZ9Fiv1V5CoWhrV 7RpGsmwVg
net::ERR_BLOCKED_BY_CLIENT
x GET https://www.google.co.in/pagead/ www.google.co.in/pag...5CoWhrV 7RpGsmwVg:1
lvz?evtid=ACd6KtzZ40h0LZgbnYYZ8gxW6c
xPBX2..inAppBootstrap%3AUnclassified&aaz=1&sigh=AB9vU40PfuZ9Fiv1V5CoWhrV 7RpGsmwVg
net::ERR_BLOCKED_BY_CLIENT
x GET chrome-extension://invalid/ net::ERR_FAILED
cast sender.js:10
x GET chrome-extension://invalid/ net::ERR_FAILED
cast sender.js:10
x GET https://static.doubleclick.net/instream/ad_status.js
net::ERR_BLOCKED_BY_CLIENT
> document.getElementById("inner-header-container").style.visibility = 'hidden'
< 'hidden'
⚠ The resource https://i.ytimg.com/generate_204 was preloaded using link featured:1
preload but not used within a few seconds from the window's load event. Please make
sure it has an appropriate 'as' value and it is preloaded intentionally.
>
```

3.4 Changes happening at Client

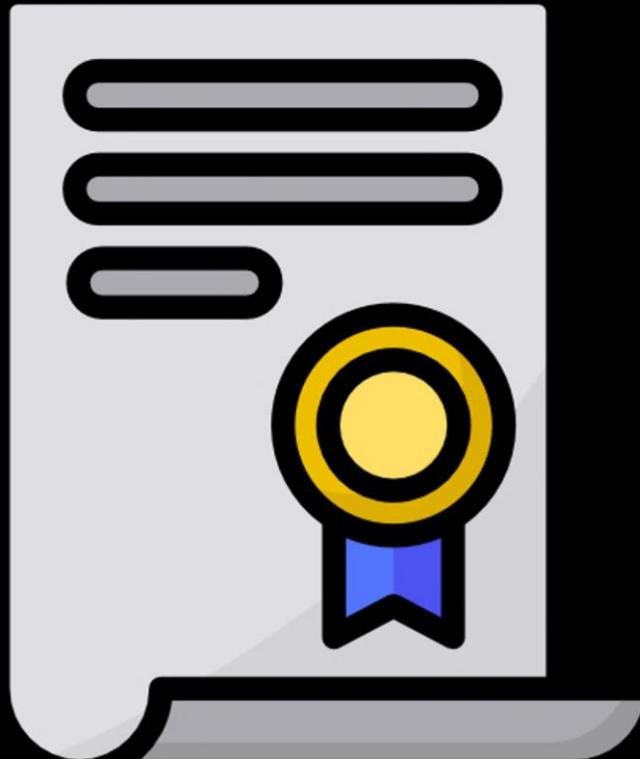
1. Changes made are temporary
2. Affect only the current session
3. Not saved to the server
4. Reset upon page reload
5. Useful for testing, not permanent fixes



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

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show source outline image report Options...

Check by css

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

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

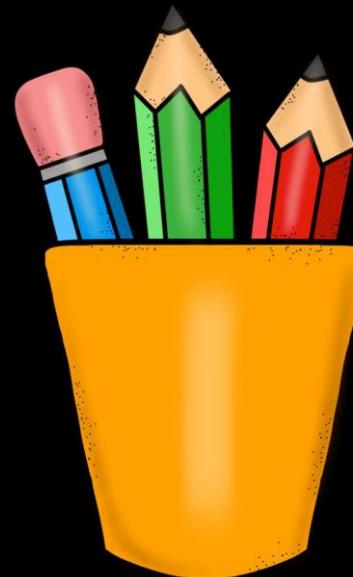
1. Error Stray start tag `html`.
From line 9, column 1 to line 9, column 6
`></body><html>`

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.



KG Coding

Some Other One shot Video Links:

- [Complete HTML](#)
- [Complete CSS](#)
- [Complete JavaScript](#)
- [Complete React and Redux](#)
- [One shot University Exam Series](#)



<http://www.kgcoding.in/>

Our YouTube Channels

KG Coding Android App



[KG Coding](#)



[Knowledge GATE](#)



[KG Placement Prep](#)



[Sanchit Socket](#)

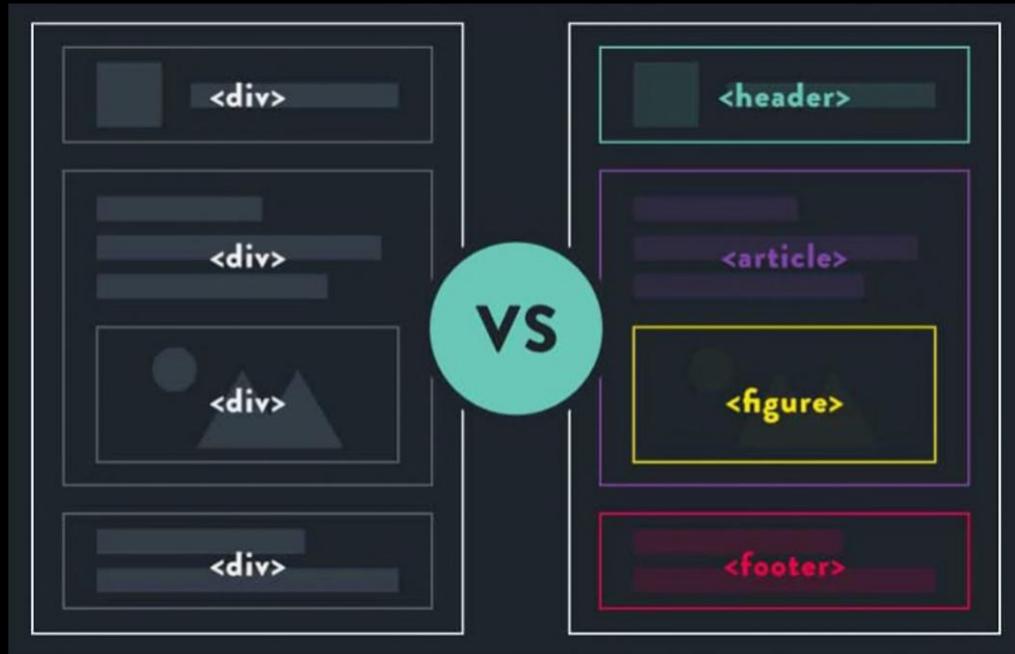
Level 4

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: `<div>`, ``, `<i>`, ``.

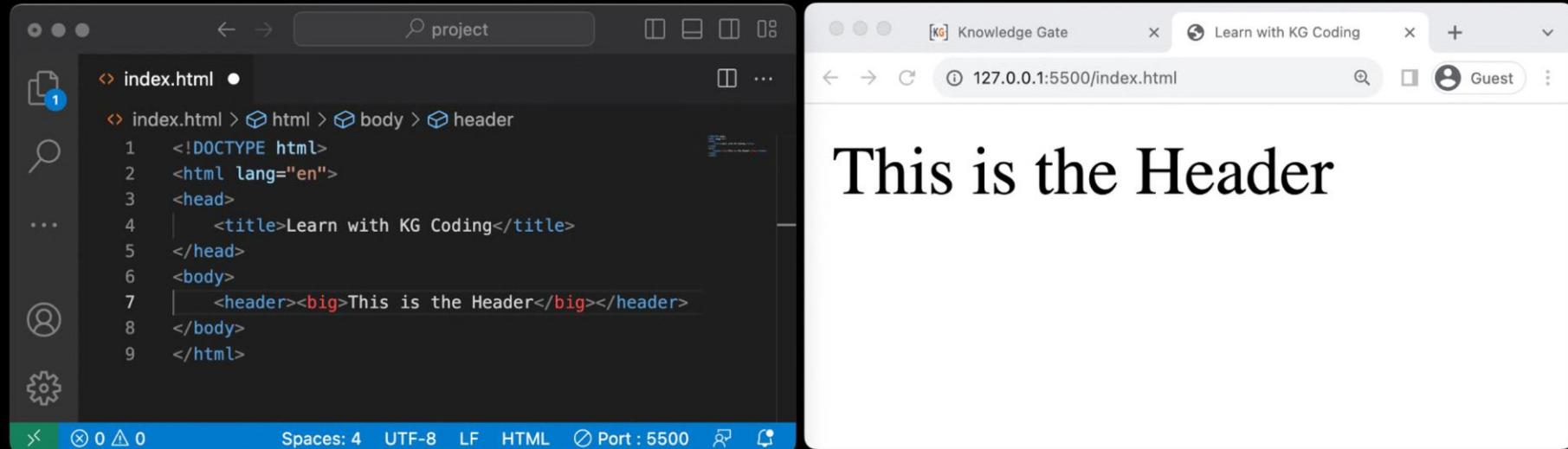
Level 4

HTML and Project Structure



2. Body Tags

2.1 Header Tag



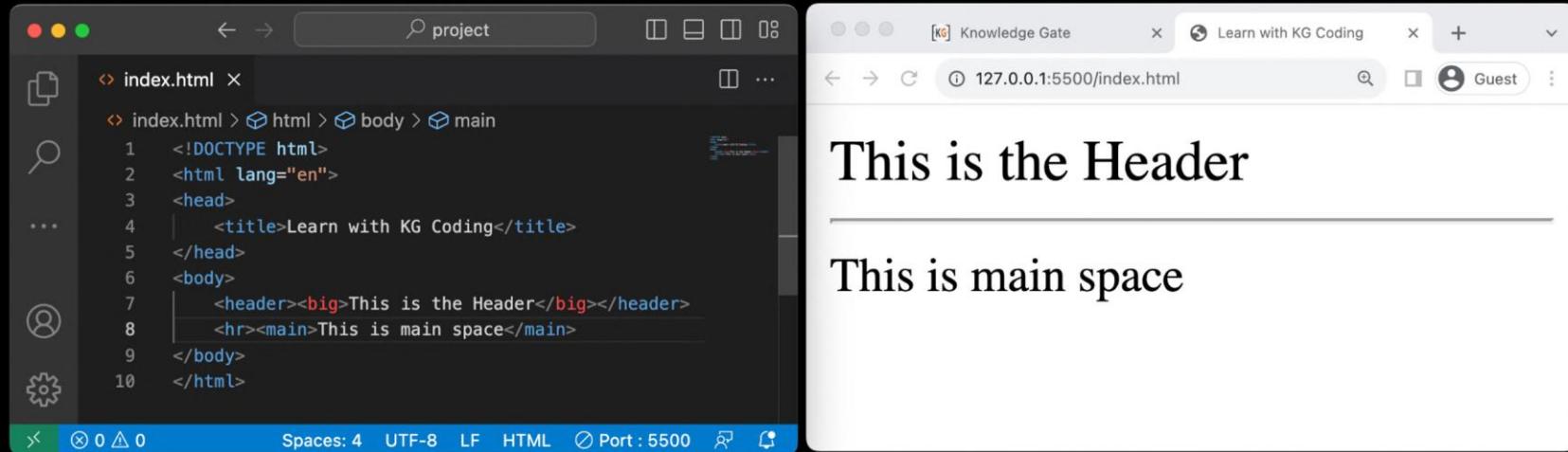
The image shows a code editor on the left and a browser window on the right. The code editor displays the file 'index.html' with the following content:

```
<!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" displayed in a large, bold black font.

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



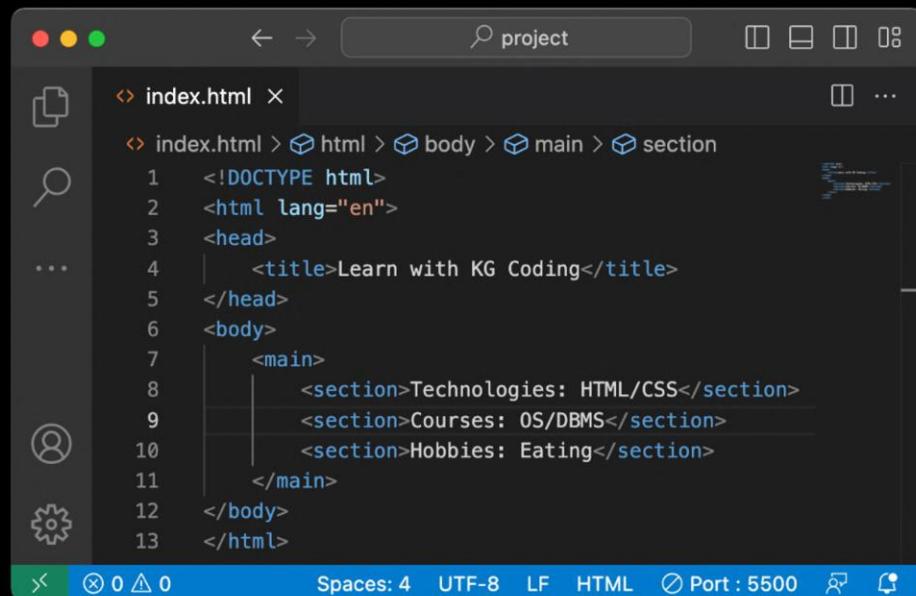
The image shows a code editor on the left and a browser window on the right. The code editor displays the file 'index.html' with the following content:

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

The browser window shows the rendered HTML. The header section contains the text "This is the Header" in a large, bold font. Below it is a horizontal line (hr). The main content area contains the text "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 like site navigation or footer.

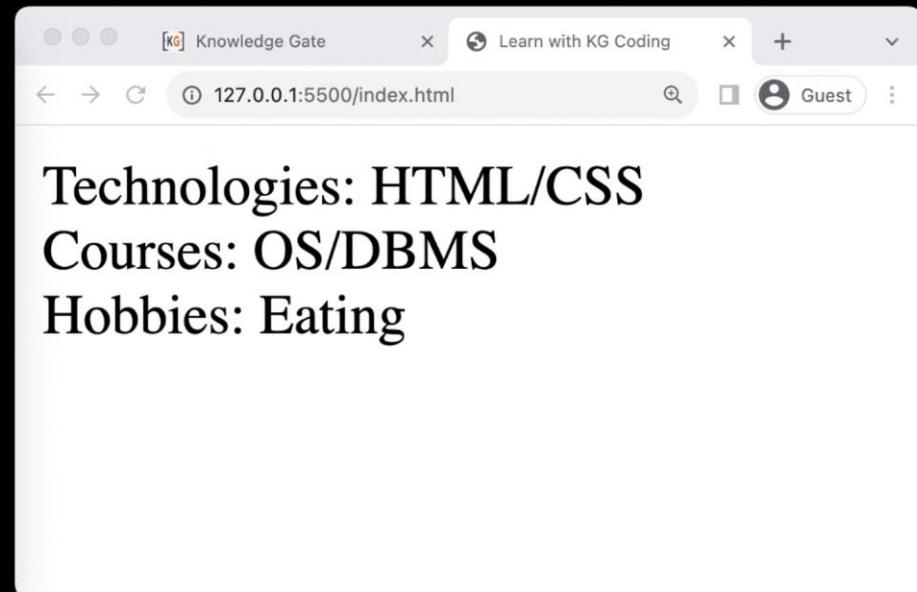
2.2.1 Section Tag



A screenshot of a code editor window titled "index.html". The code is as follows:

```
<!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>
```

The code editor interface includes a sidebar with icons for file, search, and settings, and a bottom bar with status indicators.

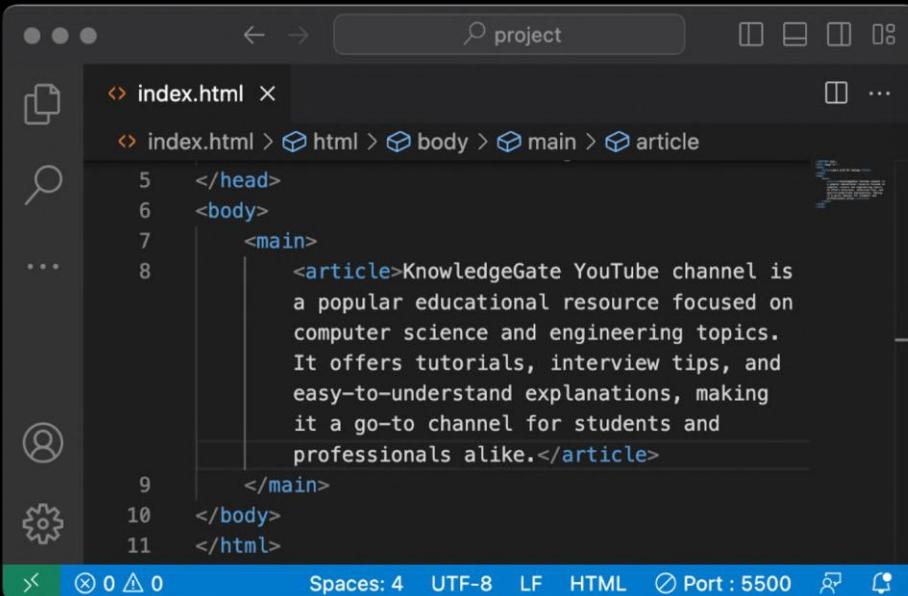


A screenshot of a web browser window titled "Learn with KG Coding". The URL is "127.0.0.1:5500/index.html". The rendered content is:

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

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

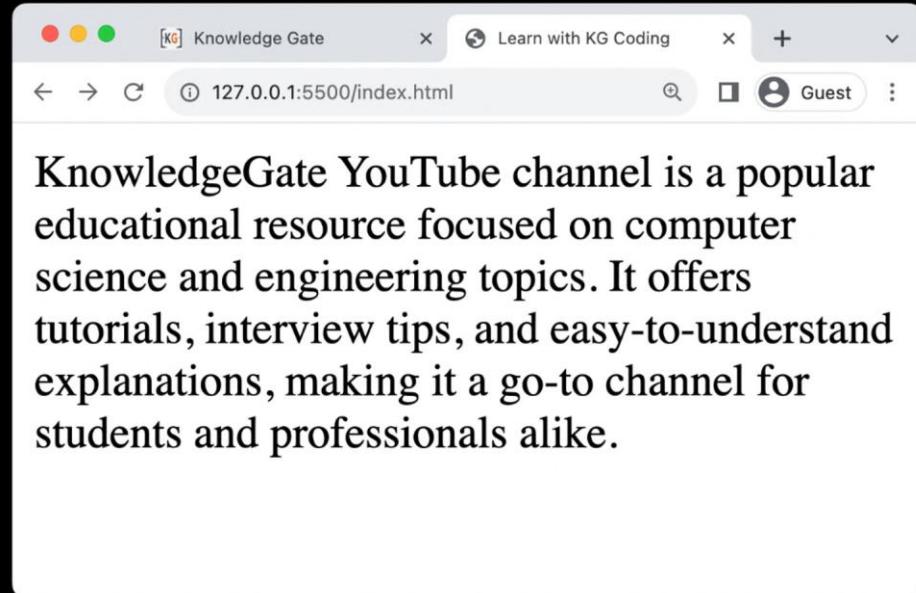
2.2.2 Article Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following HTML code:

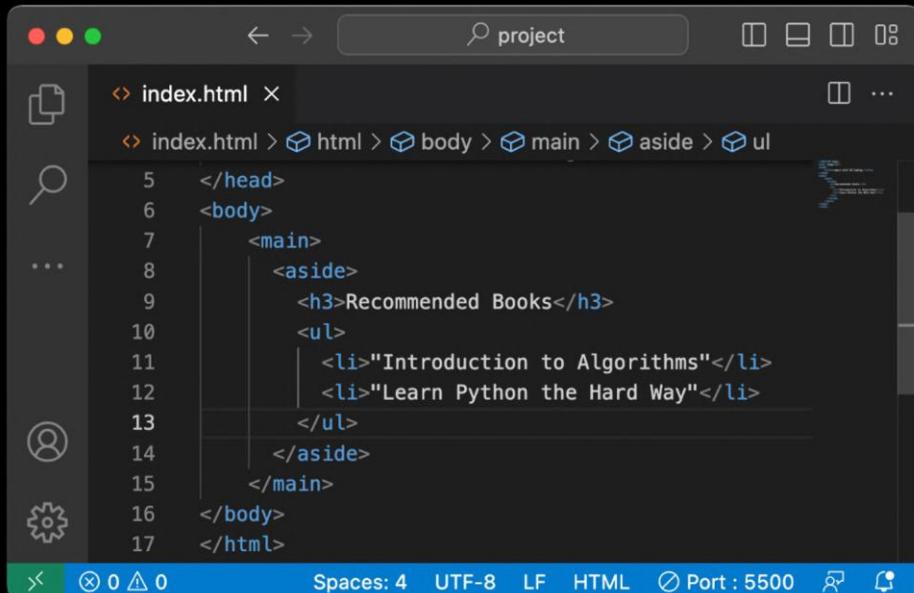
```
<head>
<body>
  <main>
    <article>KnowledgeGate YouTube channel is a popular educational resource focused on computer science and engineering topics. It offers tutorials, interview tips, and easy-to-understand explanations, making it a go-to channel for students and professionals alike.</article>
  </main>
</body>
</html>
```

The "article" tag is highlighted with a blue selection bar. The status bar at the bottom shows "Spaces: 4" and "Port : 5500".



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

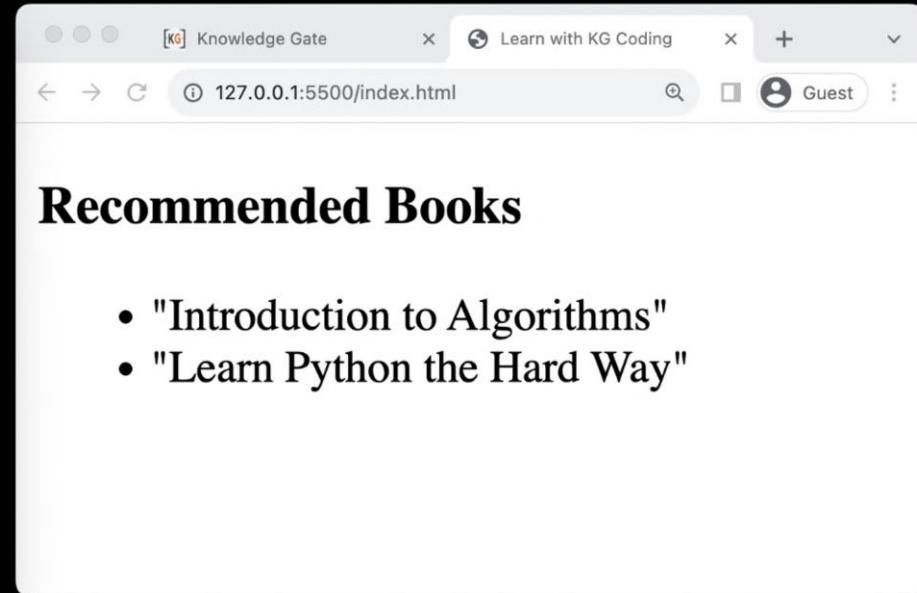
2.2.3 Aside Tag



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following HTML code:

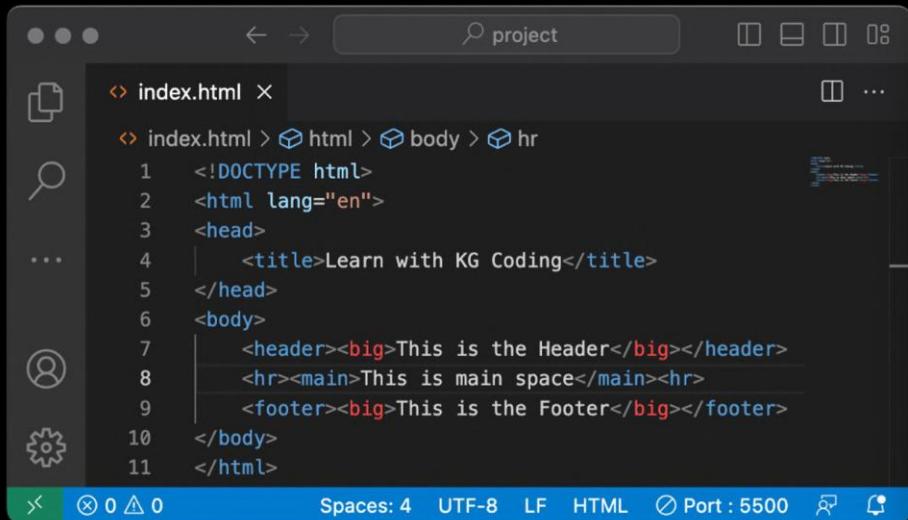
```
</head>
<body>
  <main>
    <aside>
      <h3>Recommended Books</h3>
      <ul>
        <li>"Introduction to Algorithms"</li>
        <li>"Learn Python the Hard Way"</li>
      </ul>
    </aside>
  </main>
</body>
</html>
```

The code editor has a dark theme with light-colored syntax highlighting. A sidebar on the left contains icons for file operations like new, open, save, and search.



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**.

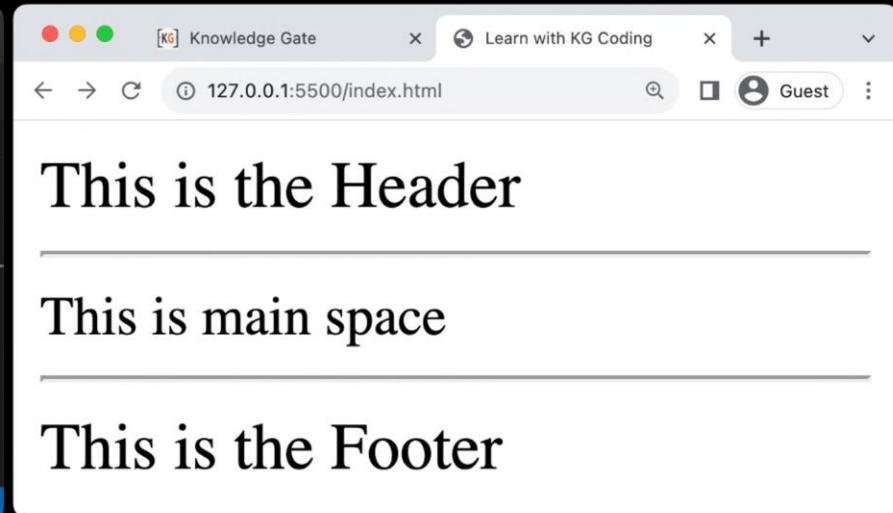
2.3 Footer Tag



A screenshot of a code editor window titled "index.html". The code is structured as follows:

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

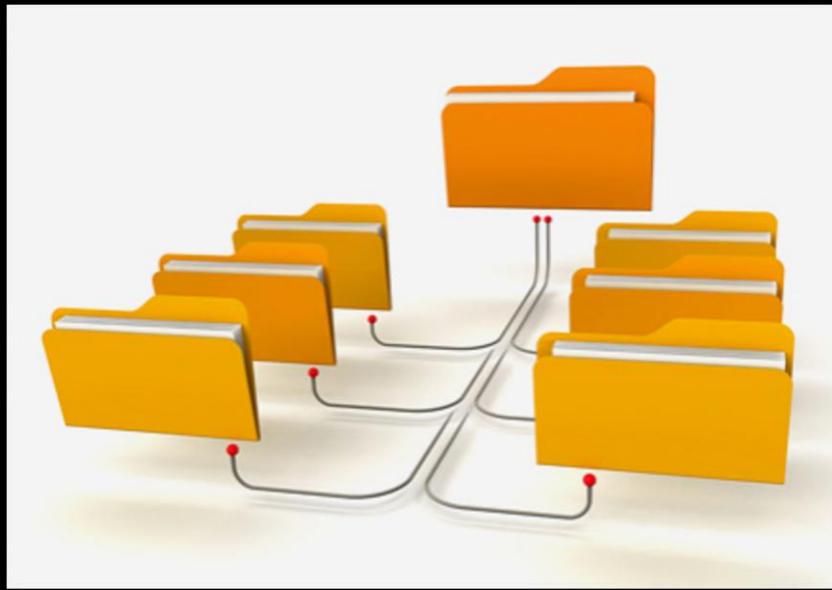
The editor interface includes a sidebar with icons for file operations, search, and user profile, and a bottom bar with tabs for "Spaces: 4", "UTF-8", "LF", "HTML", "Port: 5500", and other developer tools.



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.

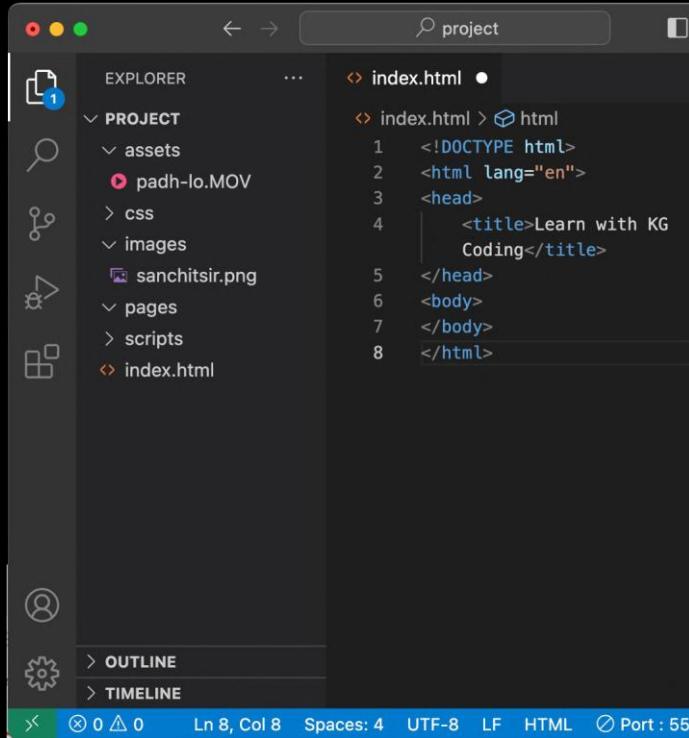
Level 4

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.

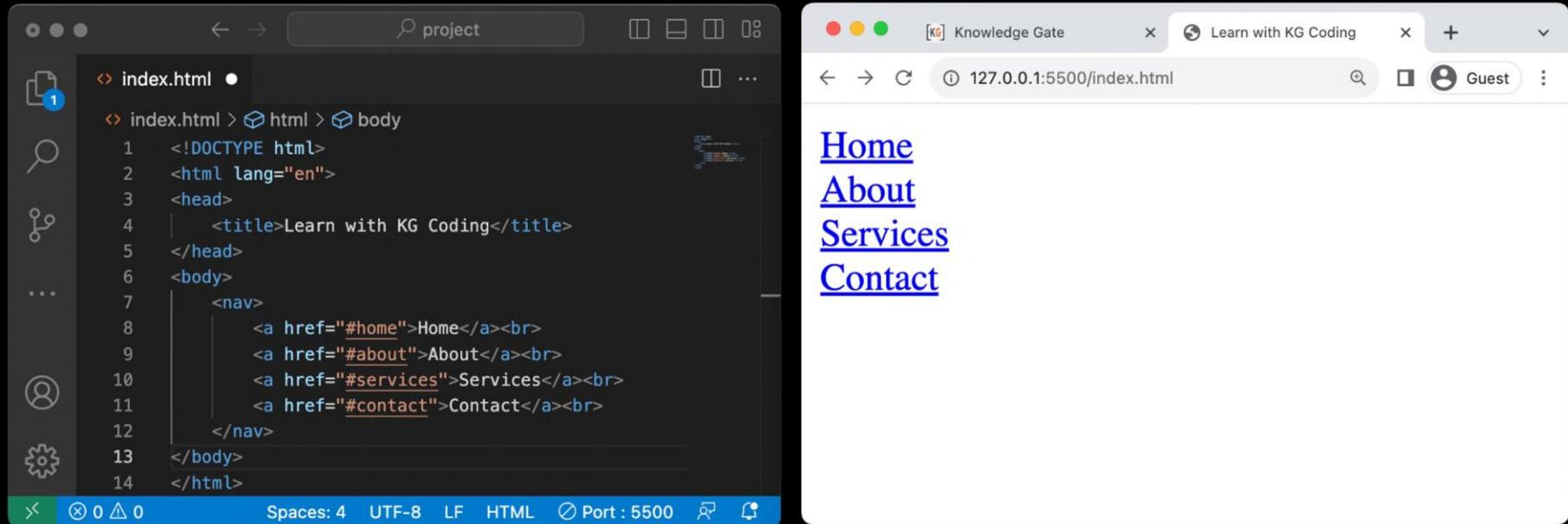
Level 4

HTML and Project Structure



4. More Tags

4.1 Navigation Tags



The image shows a code editor on the left and a web browser 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>
    <nav>
        <a href="#">Home</a><br>
        <a href="#">About</a><br>
        <a href="#">Services</a><br>
        <a href="#">Contact</a><br>
    </nav>
</body>
</html>
```

The browser window shows the rendered HTML with five blue underlined links: Home, About, Services, Contact, and another Home link.

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 ``, `` 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:** <div>, <p>, <h1>, , .

Inline Elements

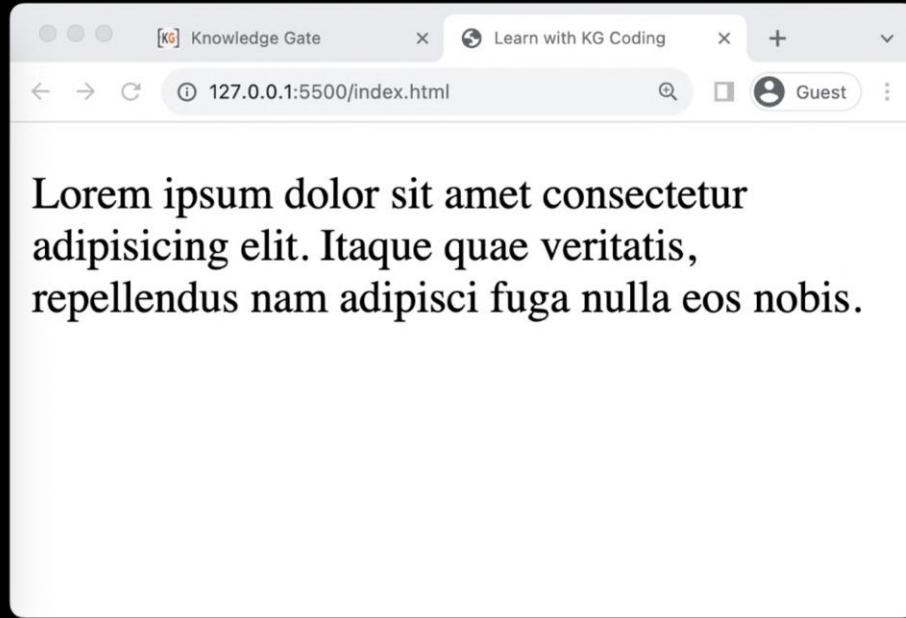
- **Flow:** Stay in line with text.
- **Width:** Just as wide as the content.
- **No Break:** No new line between elements.
- **Limited Styling:** Can't set size easily.
- **Examples:** , <a>, , , .

4.3 Div Tags

The screenshot shows a code editor interface with the following details:

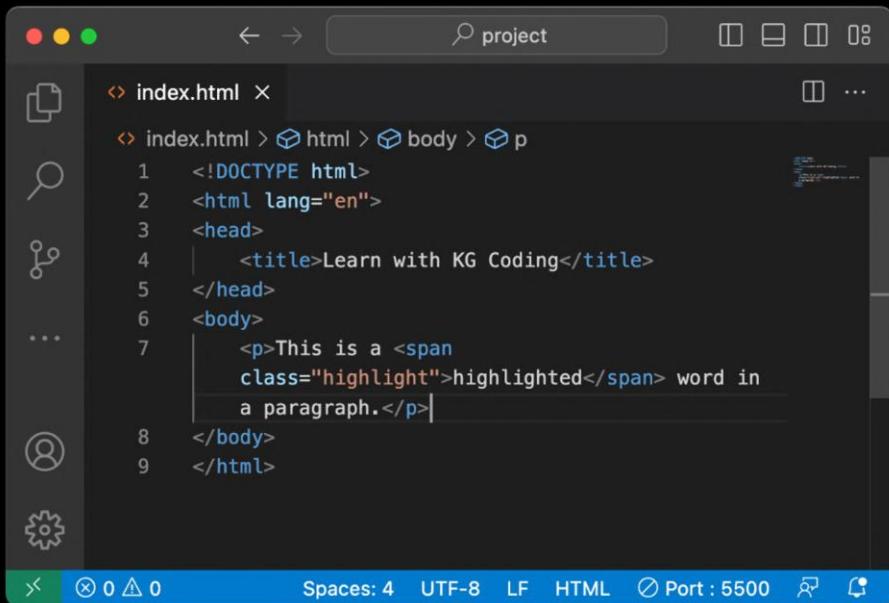
- Header:** A search bar with the placeholder "project".
- Sidebar:** On the left, there are several icons: a file icon, a magnifying glass, a circular progress bar, three dots, a user icon, and a gear icon.
- Code Area:** The main area displays the following HTML code:

```
<head>
  <title>Learn with KG Coding</title>
</head>
<body>
  <div>
    <p> Lorem ipsum dolor sit amet
    consectetur adipisicing elit. Itaque quae
    veritatis, repellendus nam adipisci fuga
    nulla eos nobis.</p>
  </div>
</body>
</html>
```
- Bottom Status Bar:** Shows "Spaces: 4", "UTF-8", "LF", "HTML", "Port: 5500", and two small icons.



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.

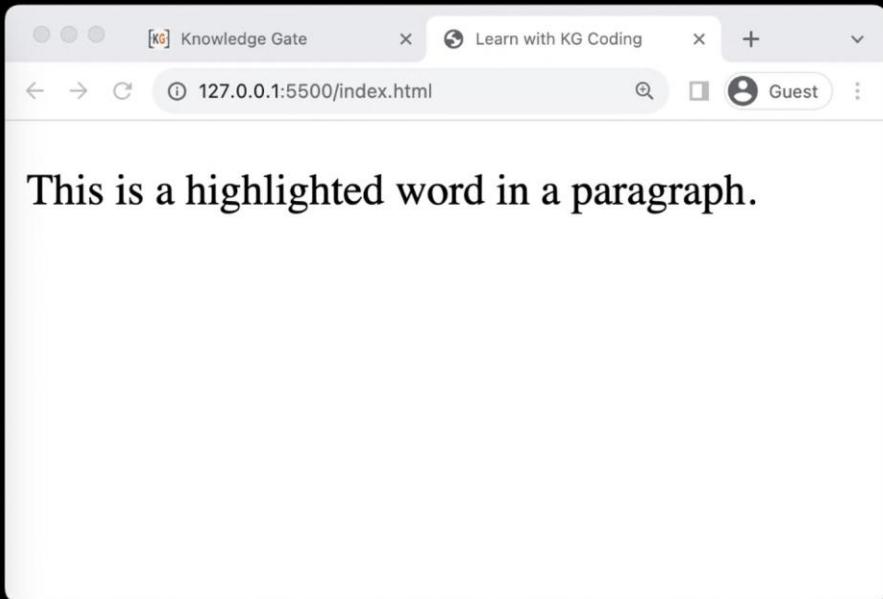
4.4 Span Tags



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <p>This is a <span class="highlight">highlighted</span> word in
    a paragraph.</p>
</body>
</html>
```

The word "highlighted" is highlighted with a blue background and white text. The status bar at the bottom shows "Spaces: 4" and "Port : 5500".



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

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

```
<!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 on the right shows the rendered output of the HTML code, which is an ordered list:

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.

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"`

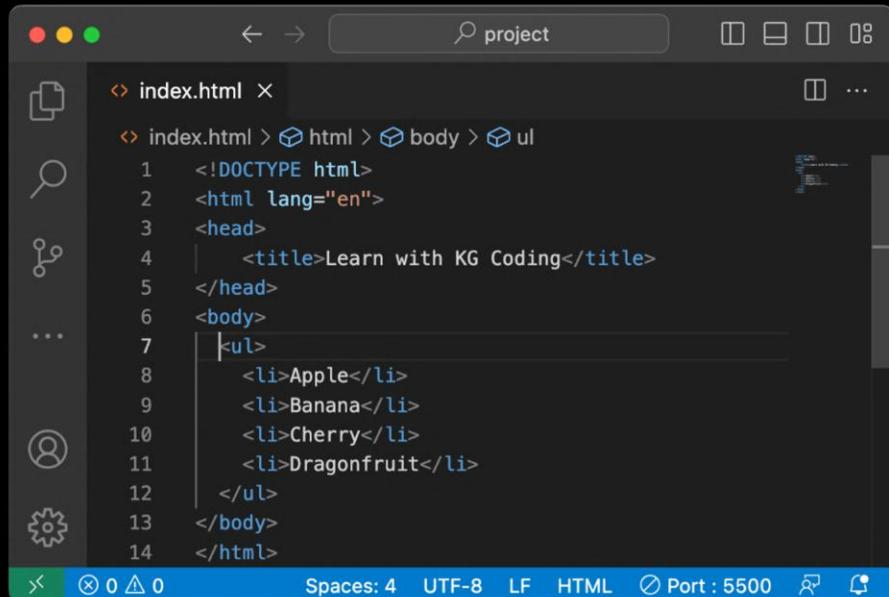
A. Apple
B. Banana
C. Cherry
D. Dragonfruit

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

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

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

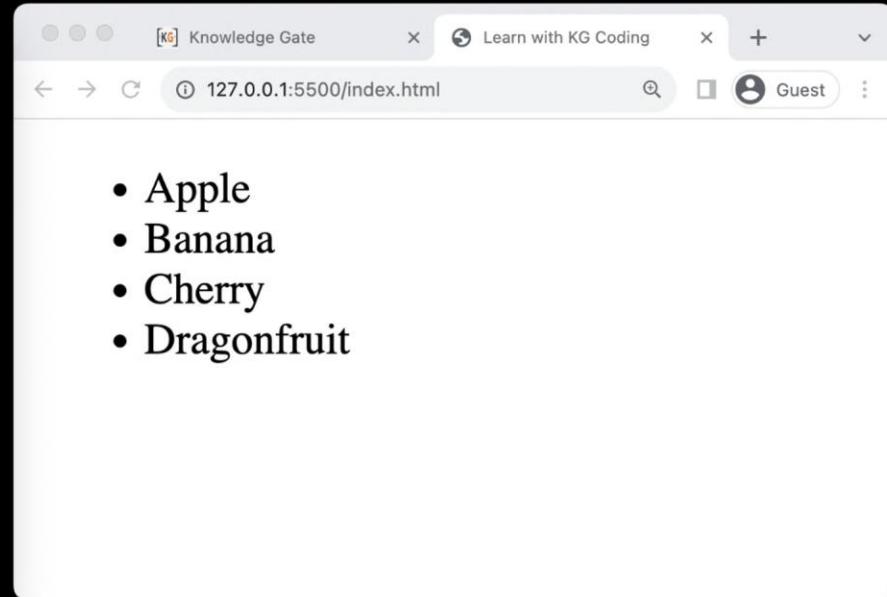
1.3 Unordered Lists



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Learn with KG Coding</title>
5  </head>
6  <body>
7  |<ul>
8  |   <li>Apple</li>
9  |   <li>Banana</li>
10 |   <li>Cherry</li>
11 |   <li>Dragonfruit</li>
12 |</ul>
13 </body>
14 </html>
```

The code editor has a dark theme with light-colored syntax highlighting. A sidebar on the left contains various icons for file operations like copy, paste, search, and settings.



A screenshot of a web browser window titled "Knowledge Gate". The address bar shows "127.0.0.1:5500/index.html". The page content is:

- Apple
- Banana
- Cherry
- 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.

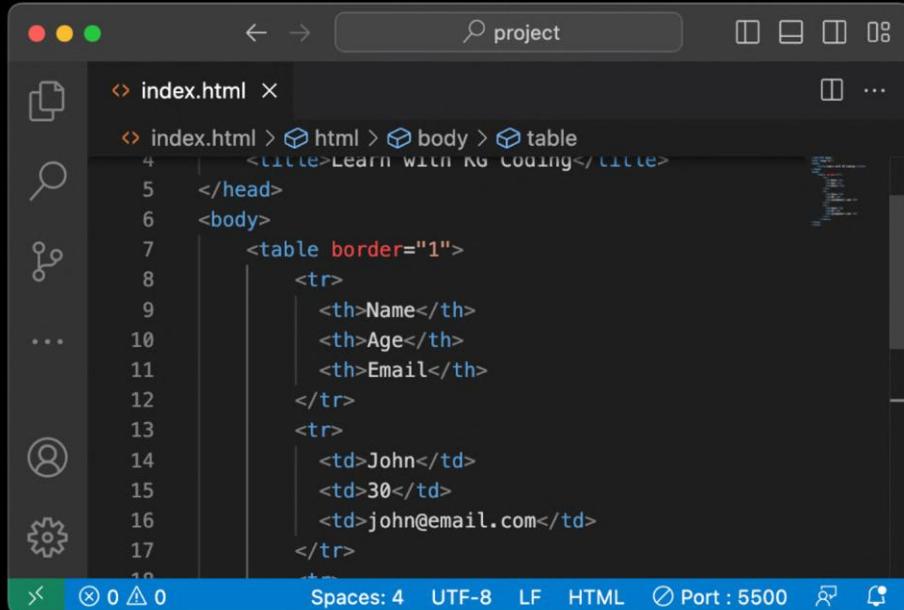
Level 5

List, Tables & Forms



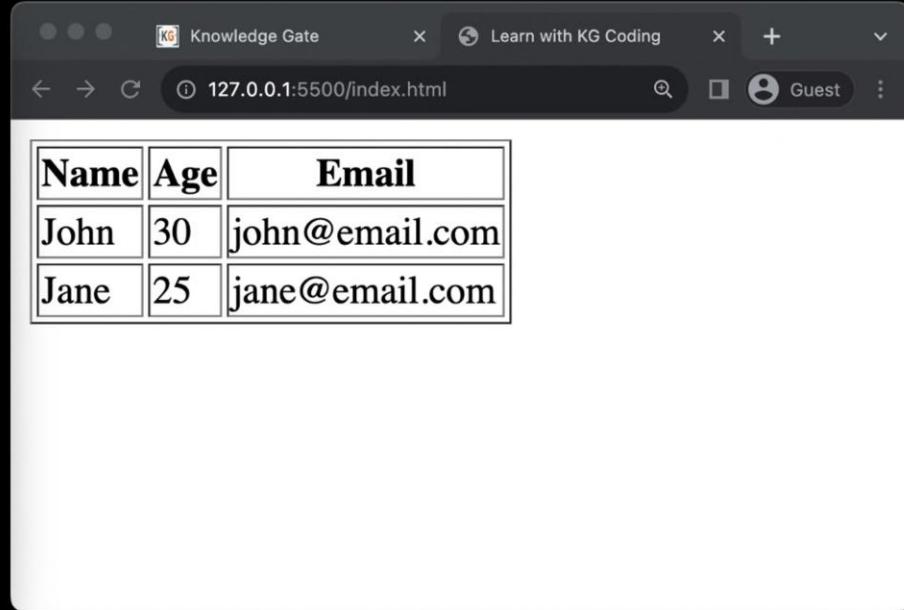
2. Table
Tag

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



The screenshot shows a code editor window with the file 'index.html' open. The code defines a simple table with three columns: Name, Age, and Email. It consists of two header rows (tr) and two data rows (tr). The header row contains three th tags, and the data row contains three td tags. The code is written in a dark-themed editor.

```
<!DOCTYPE html>
<html>
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>Email</th>
        </tr>
        <tr>
            <td>John</td>
            <td>30</td>
            <td>john@email.com</td>
        </tr>
        <tr>
            <td>Jane</td>
            <td>25</td>
            <td>jane@email.com</td>
        </tr>
    </table>
</body>
</html>
```

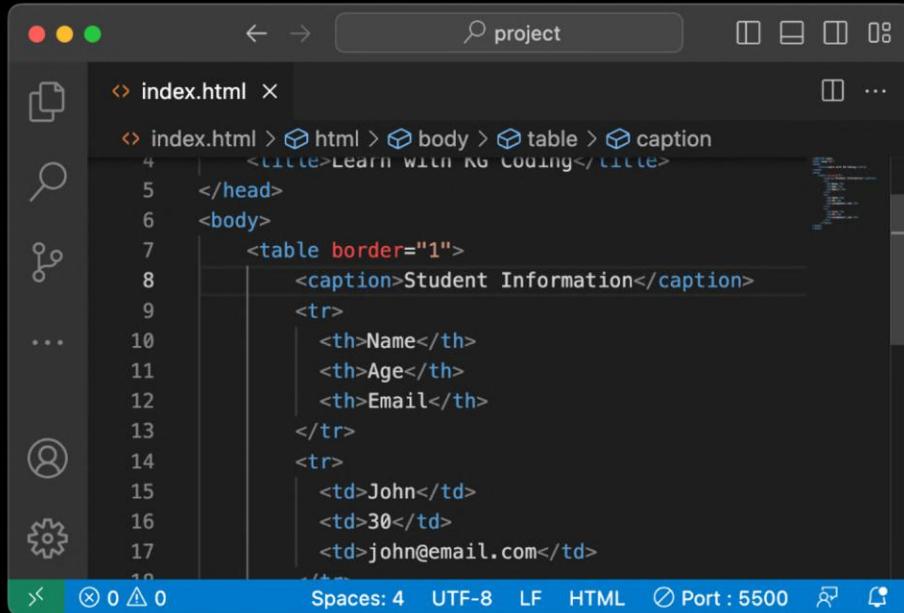


The screenshot shows a web browser window titled 'Knowledge Gate' with the URL '127.0.0.1:5500/index.html'. The page displays a table with three columns: Name, Age, and Email. There are two rows of data: one for John (30, john@email.com) and one for Jane (25, jane@email.com). The table has a border of 1.

Name	Age	Email
John	30	john@email.com
Jane	25	jane@email.com

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.

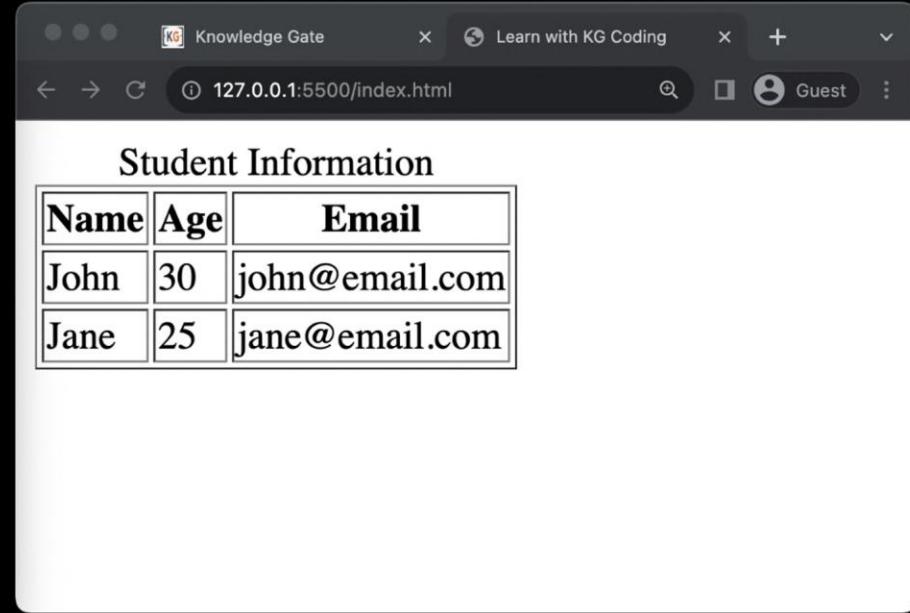
2.2 Captions



The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar with various icons. The main area displays the following HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Learn HTML by Coding</title>
  </head>
  <body>
    <table border="1">
      <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>
    </table>
  </body>
</html>
```

The code editor has a status bar at the bottom with the following information: Spaces: 4, UTF-8, LF, HTML, Port: 5500.



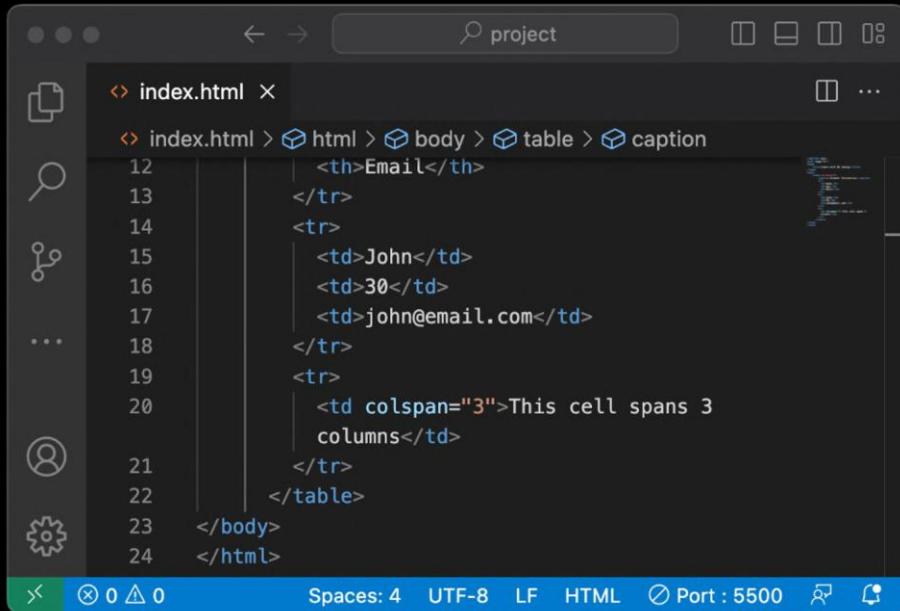
The screenshot shows a web browser window titled "Knowledge Gate" with the URL "127.0.0.1:5500/index.html". The page content is:

Student Information

Name	Age	Email
John	30	john@email.com
Jane	25	jane@email.com

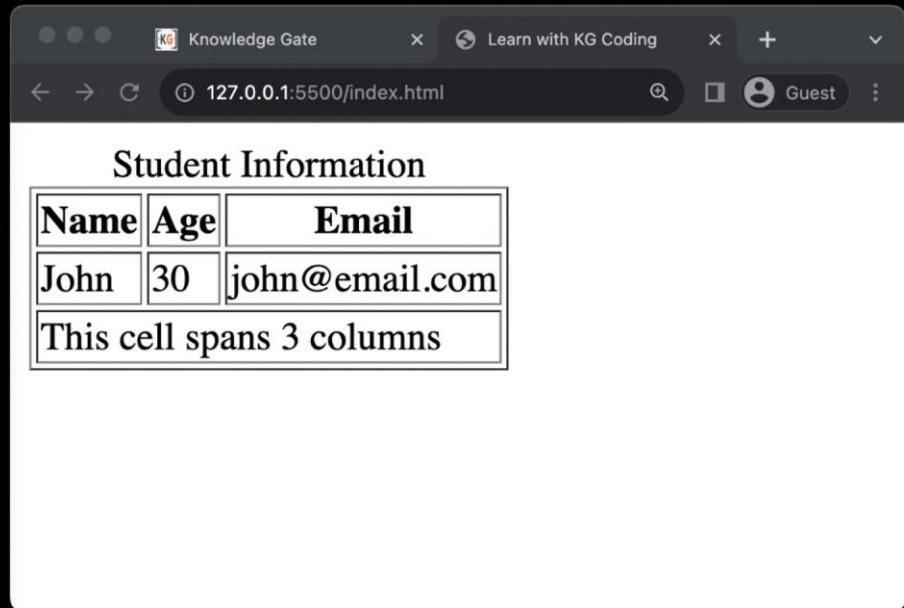
1. **Purpose:** Provides a title or description for a table.
2. **Placement:** Must be inserted **immediately after** the `<table>` opening tag.
3. **Alignment:** **Centered** above the table by default.
4. **Accessibility:** Helps screen readers understand the table's purpose.

2.3 Col Spans



A screenshot of a code editor window titled "index.html". The code displays a simple HTML table with three columns: Name, Age, and Email. The first two rows have three data cells each. The third row contains a single data cell with the text "This cell spans 3 columns" and a colspan="3" attribute.

```
<th>Email</th>
</tr>
<tr>
  <td>John</td>
  <td>30</td>
  <td>john@email.com</td>
</tr>
<tr>
  <td colspan="3">This cell spans 3
    columns</td>
</tr>
</table>
</body>
</html>
```



A screenshot of a web browser window titled "Knowledge Gate" showing the URL "127.0.0.1:5500/index.html". The page title is "Student Information". It features a table with three columns: Name, Age, and Email. The first two rows contain standard data. The third row contains a single cell with the text "This cell spans 3 columns" and a colspan="3" attribute.

Name	Age	Email
John	30	john@email.com
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.

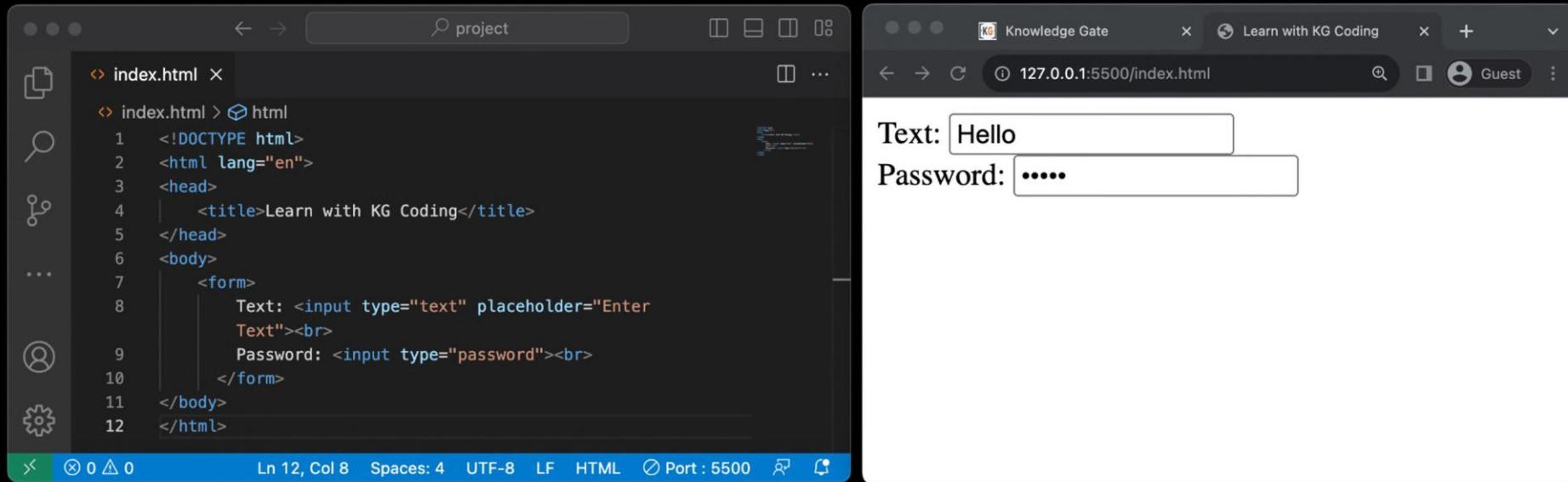
Level 5

List, Tables & Forms



3. Forms

3.1 Input Tag



The image shows a code editor on the left and a web browser 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>
    <form>
        Text: <input type="text" placeholder="Enter Text"><br>
        Password: <input type="password"><br>
    </form>
</body>
</html>
```

The browser window shows the rendered HTML. It has two input fields. The first input field is labeled 'Text:' and contains the text 'Hello'. The second input field is labeled 'Password:' and contains five dots ('.....') representing a password.

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

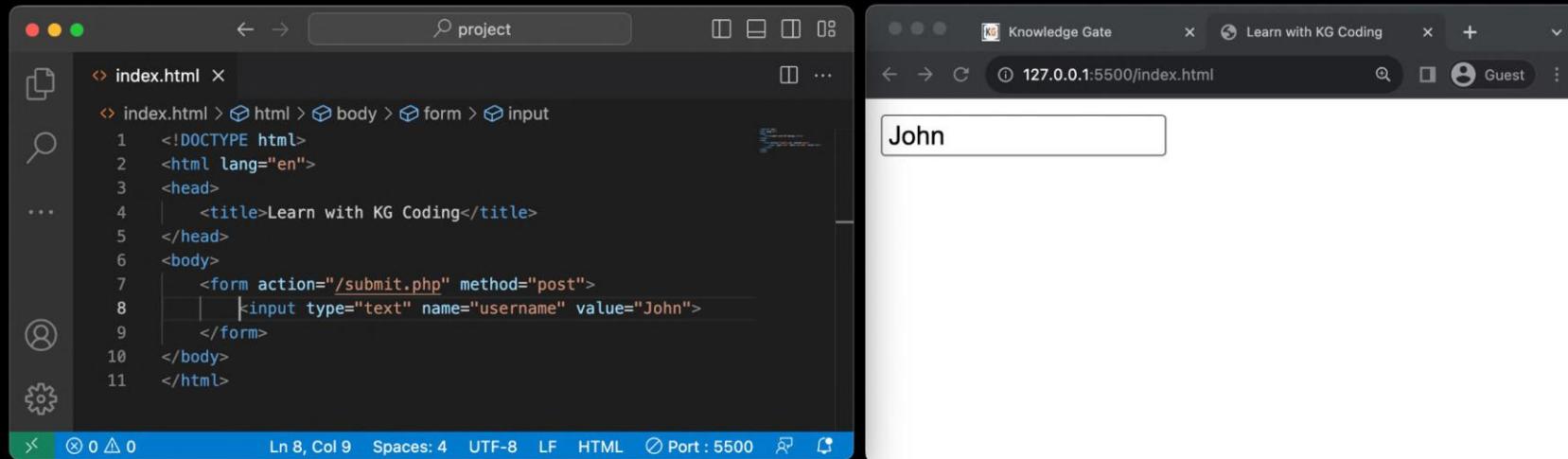
The image shows a code editor on the left and a browser window on the right. The code editor displays the file 'index.html' with the following content:

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

The browser window shows the rendered HTML with a text input field and a 'Submit' button.

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



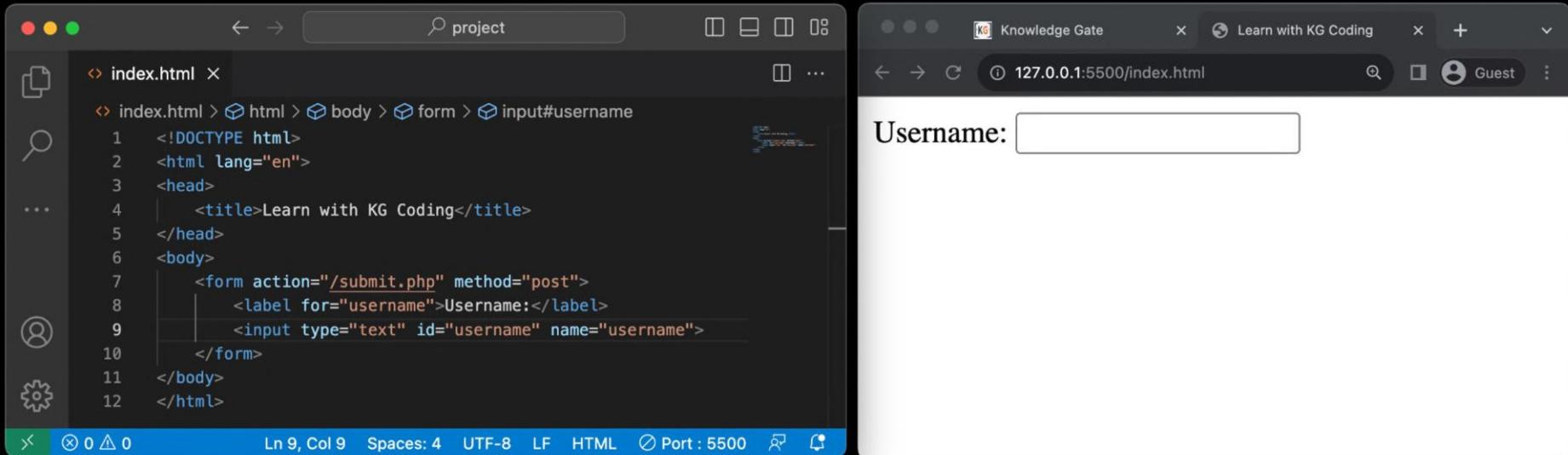
The image shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

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

The browser window shows a single 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.

3.4 Label Tag



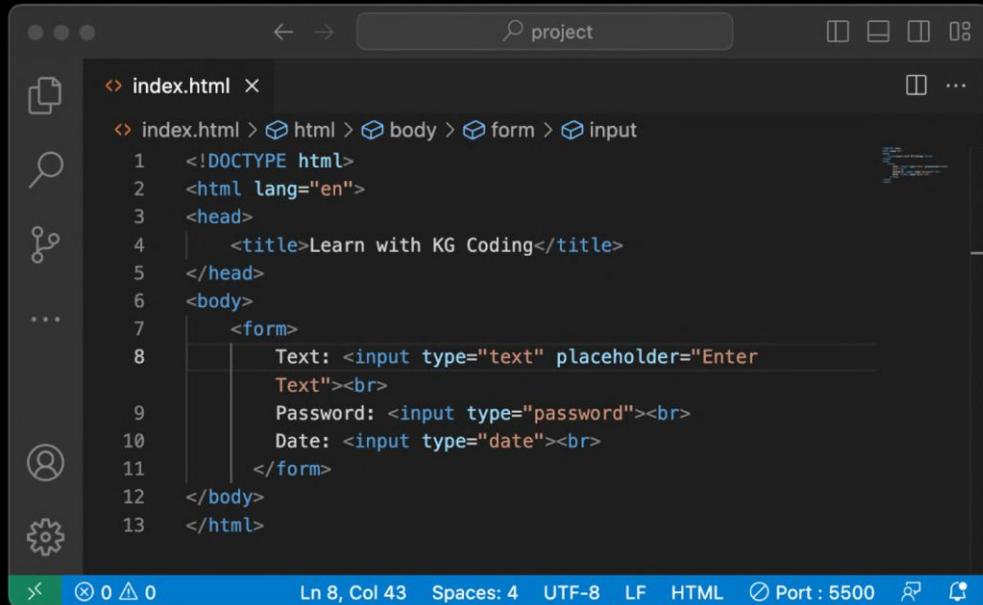
The image shows a code editor on the left and a web browser on the right. The code editor displays the file 'index.html' with the following content:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Learn with KG Coding</title>
5 </head>
6 <body>
7   <form action="/submit.php" method="post">
8     <label for="username">Username:</label>
9     <input type="text" id="username" name="username">
10   </form>
11 </body>
12 </html>
```

The browser window shows the rendered HTML with the label "Username:" followed by an empty input field.

- **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.

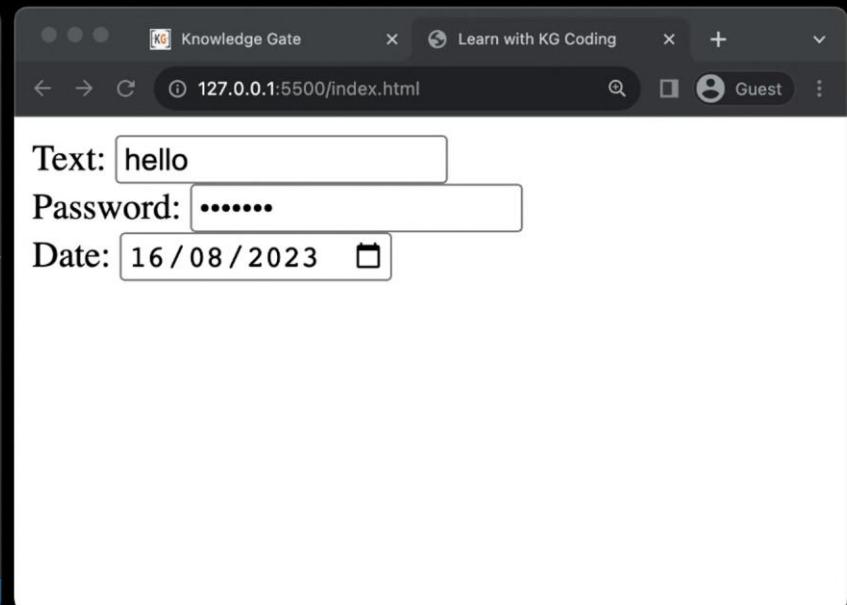
3.5 Input type: Date



A screenshot of a code editor window titled "index.html". The code editor shows the following HTML structure:

```
<!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>
```

The code editor interface includes a sidebar with various icons (file, search, copy, etc.) and a status bar at the bottom with information like "Ln 8, Col 43" and "Port : 5500".

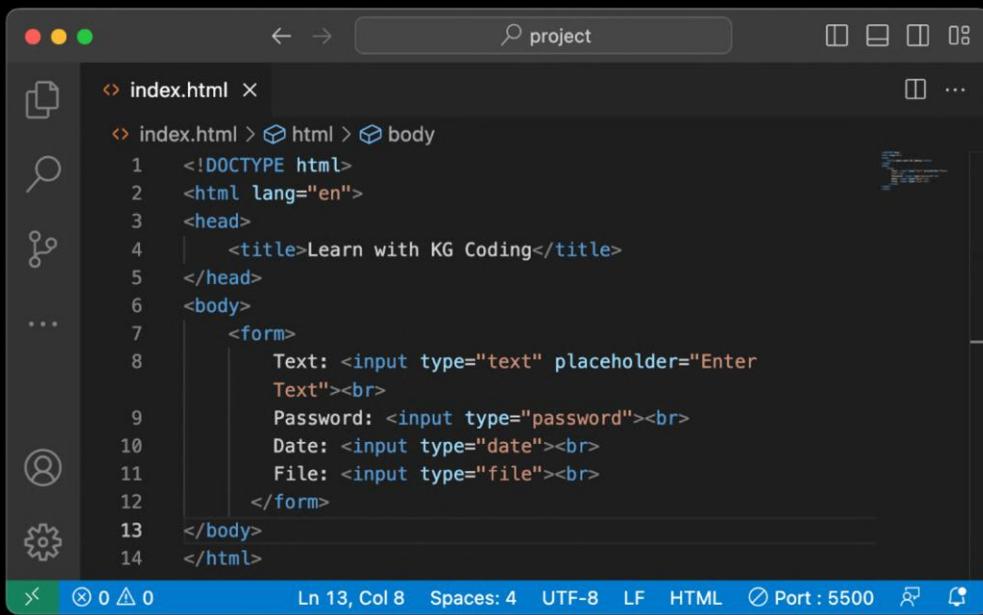


A screenshot of a web browser window titled "Knowledge Gate". The URL bar shows "127.0.0.1:5500/index.html". The page displays a form with three input fields:

- Text:
- Password:
- Date:

The browser interface includes standard navigation buttons and a user profile icon.

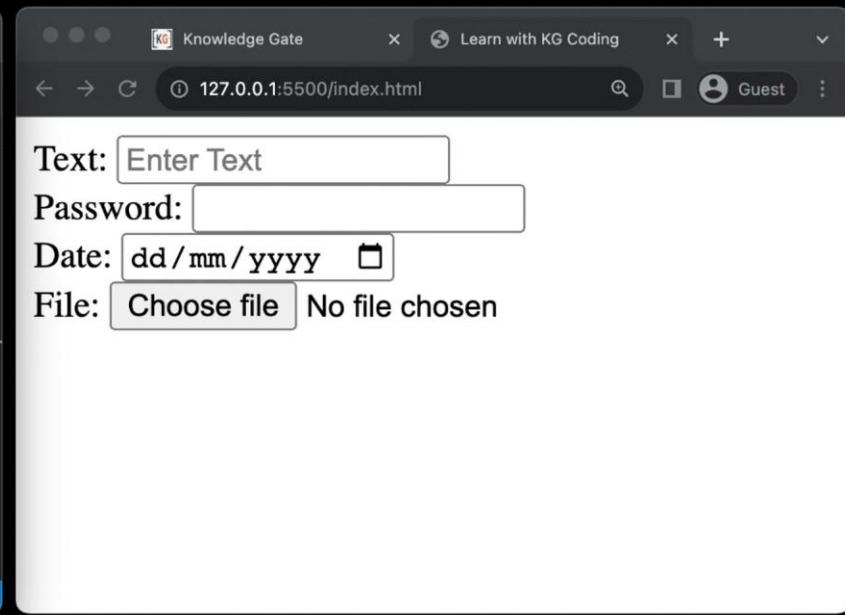
3.5 Input type: File



A screenshot of a code editor window titled "project". The file "index.html" is open, showing the following 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>
        Date: <input type="date"><br>
        File: <input type="file"><br>
    </form>
</body>
</html>
```

The code editor has a dark theme with light-colored syntax highlighting. The status bar at the bottom shows: Ln 13, Col 8, Spaces: 4, UTF-8, LF, HTML, Port : 5500.



A screenshot of a web browser window titled "Knowledge Gate". The URL is 127.0.0.1:5500/index.html. The page contains the following form fields:

Text:

Password:

Date:

File: Choose file No file chosen

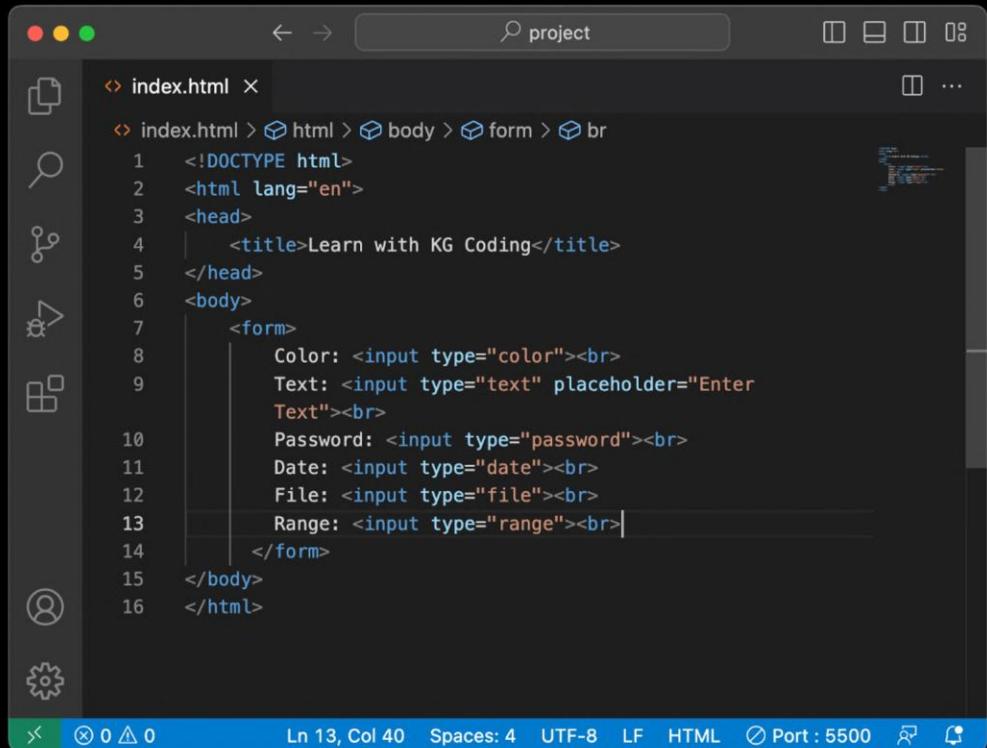
3.5 Input type: Color

The image shows a code editor on the left and a web browser 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>
        Color: <input type="color"><br>
        Text: <input type="text" placeholder="Enter Text"><br>
        Password: <input type="password"><br>
        Date: <input type="date"><br>
        File: <input type="file"><br>
    </form>
</body>
</html>
```

The browser window shows the rendered HTML. It includes five input fields: a color picker, a text input field with placeholder "Enter Text", a password input field, a date input field, and a file input field. The color picker is currently set to black. Below the color picker are three numerical input fields for Red (R), Green (G), and Blue (B), all currently set to 0.

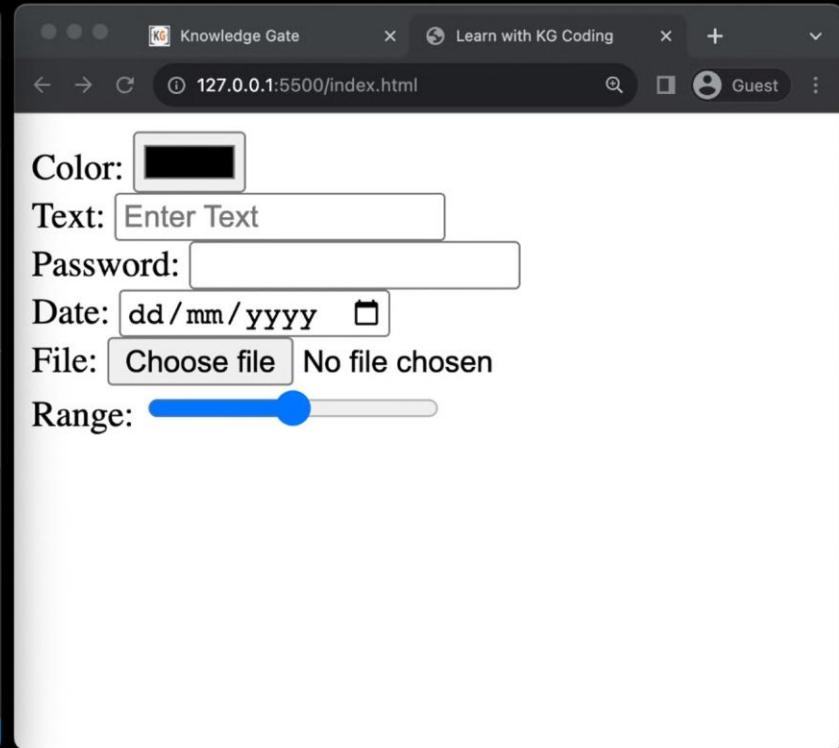
3.5 Input type: Range



A screenshot of a code editor showing the file `index.html`. The code defines a form with several input fields:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form>
        Color: <input type="color"><br>
        Text: <input type="text" placeholder="Enter Text"><br>
        Password: <input type="password"><br>
        Date: <input type="date"><br>
        File: <input type="file"><br>
        Range: <input type="range"><br>
    </form>
</body>
</html>
```

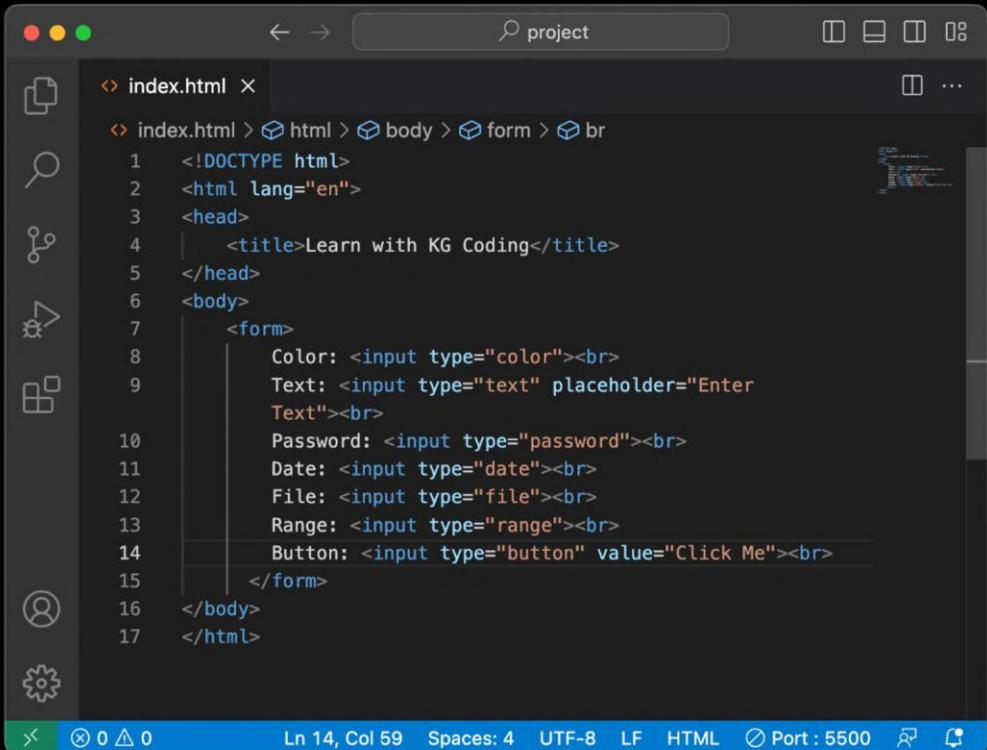
The code editor interface includes a sidebar with icons for file operations, a search bar, and status indicators at the bottom.



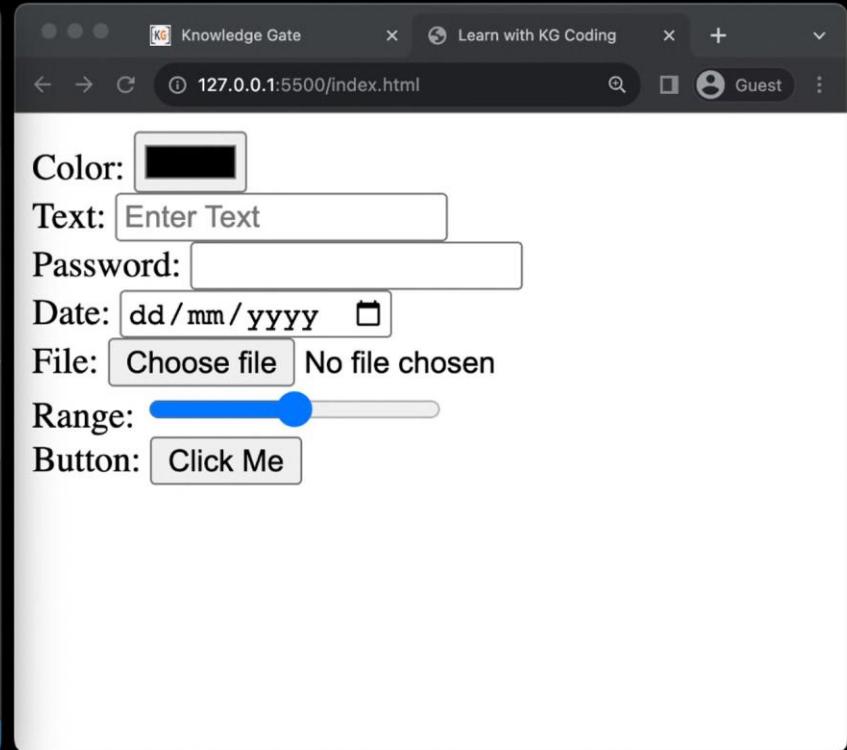
A screenshot of a web browser showing the rendered HTML from the code editor. The page displays the following inputs:

- Color: A color picker button.
- Text: An input field with placeholder text "Enter Text".
- Password: An input field of type password.
- Date: An input field of type date with a calendar icon.
- File: An input field of type file showing "Choose file" and "No file chosen".
- Range: A range slider with a blue handle and a horizontal track.

3.5 Input type: Button



```
index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form>
        Color: <input type="color"><br>
        Text: <input type="text" placeholder="Enter Text"><br>
        Password: <input type="password"><br>
        Date: <input type="date"><br>
        File: <input type="file"><br>
        Range: <input type="range"><br>
        Button: <input type="button" value="Click Me"><br>
    </form>
</body>
</html>
```



Color:

Text:

Password:

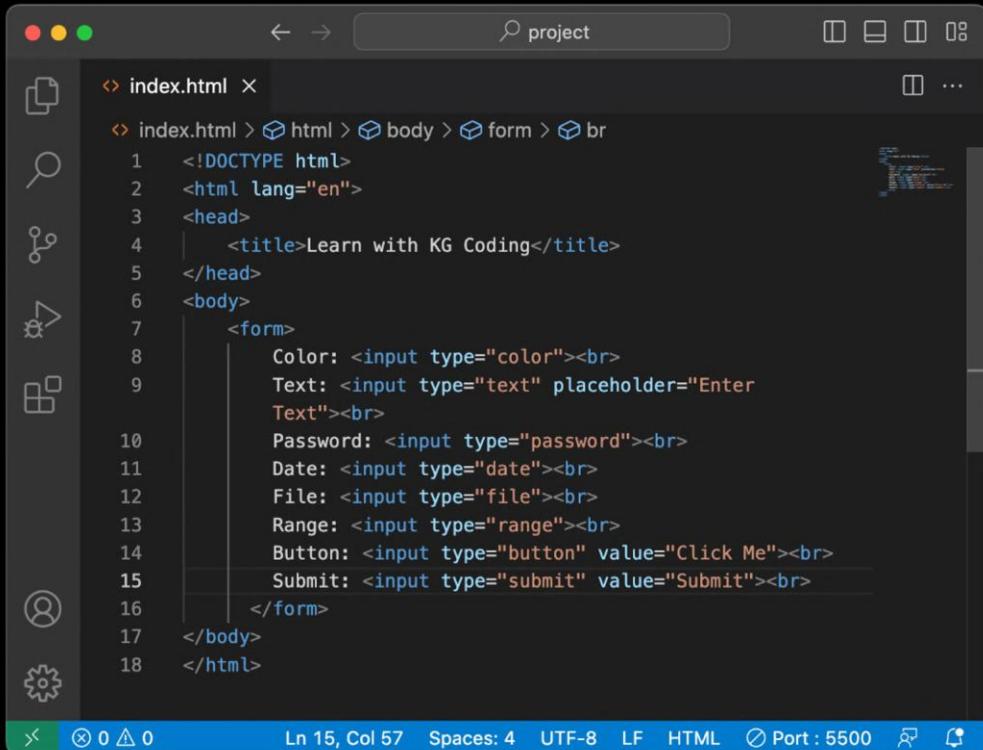
Date:

File: No file chosen

Range:

Button:

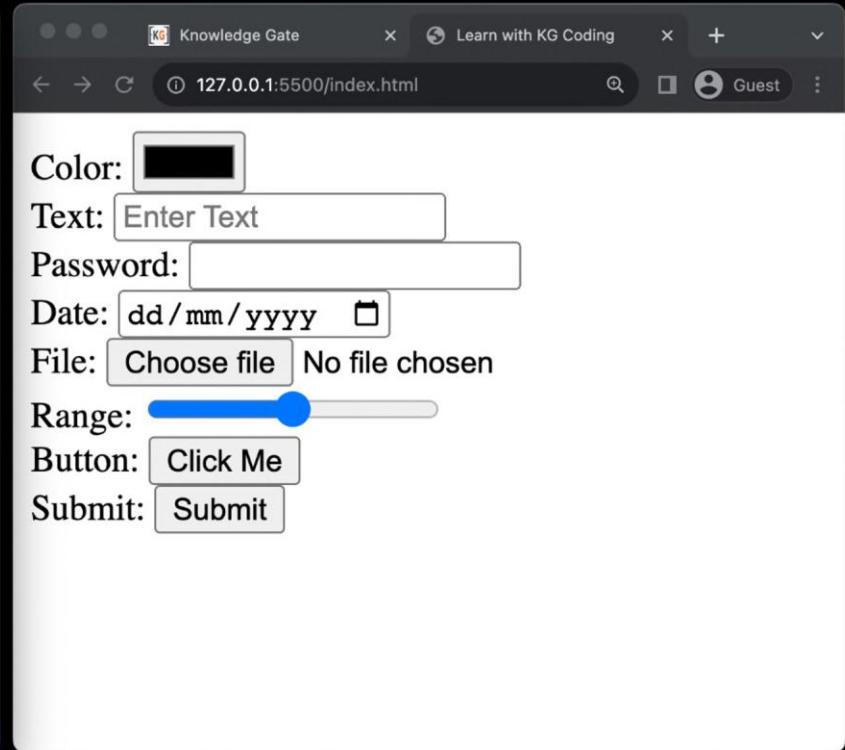
3.5 Input type: Submit



A screenshot of a code editor showing the file `index.html`. The code defines a form with several input fields:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form>
        Color: <input type="color"><br>
        Text: <input type="text" placeholder="Enter Text"><br>
        Password: <input type="password"><br>
        Date: <input type="date"><br>
        File: <input type="file"><br>
        Range: <input type="range"><br>
        Button: <input type="button" value="Click Me"><br>
        Submit: <input type="submit" value="Submit"><br>
    </form>
</body>
</html>
```

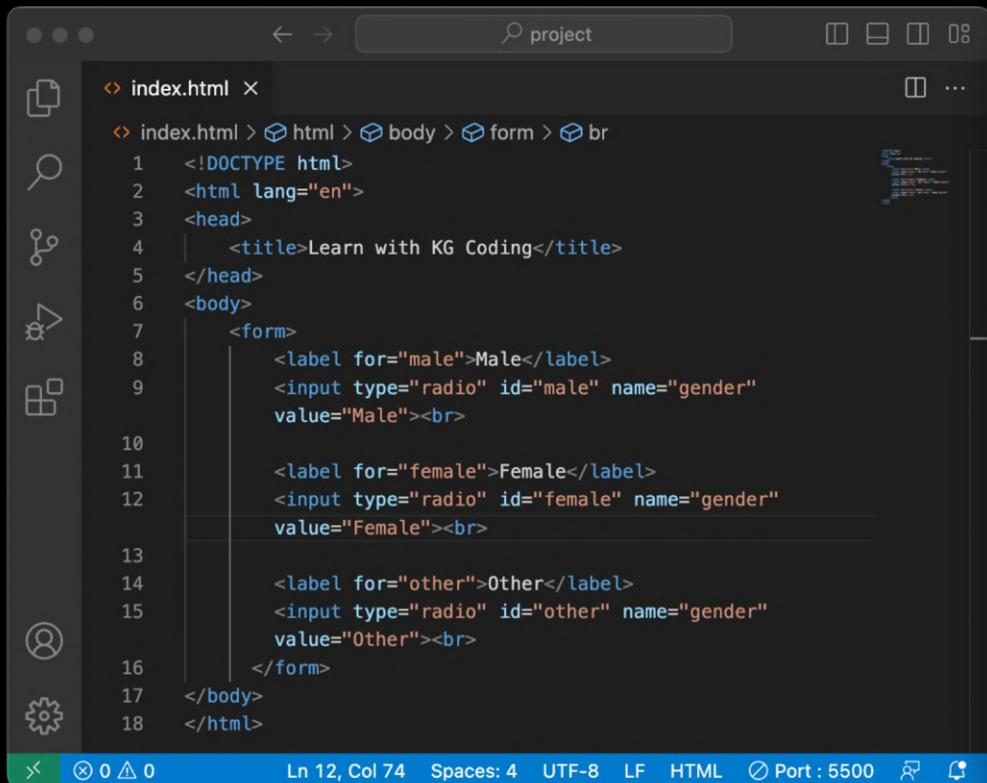
The code editor interface includes a sidebar with various icons for file operations, a search bar, and status indicators at the bottom.



A screenshot of a web browser showing the rendered form from the code editor. The browser window title is "Knowledge Gate" and the URL is "127.0.0.1:5500/index.html". The form contains the following fields:

- Color: A color picker button.
- Text: An input field with placeholder text "Enter Text".
- Password: An input field for entering a password.
- Date: A date input field with a placeholder "dd/mm/yyyy" and a calendar icon.
- File: A file input field with placeholder text "Choose file" and a message "No file chosen".
- Range: A range slider with a blue handle and a grey track.
- Button: A button labeled "Click Me".
- Submit: A button labeled "Submit".

3.5 Input type: Radio



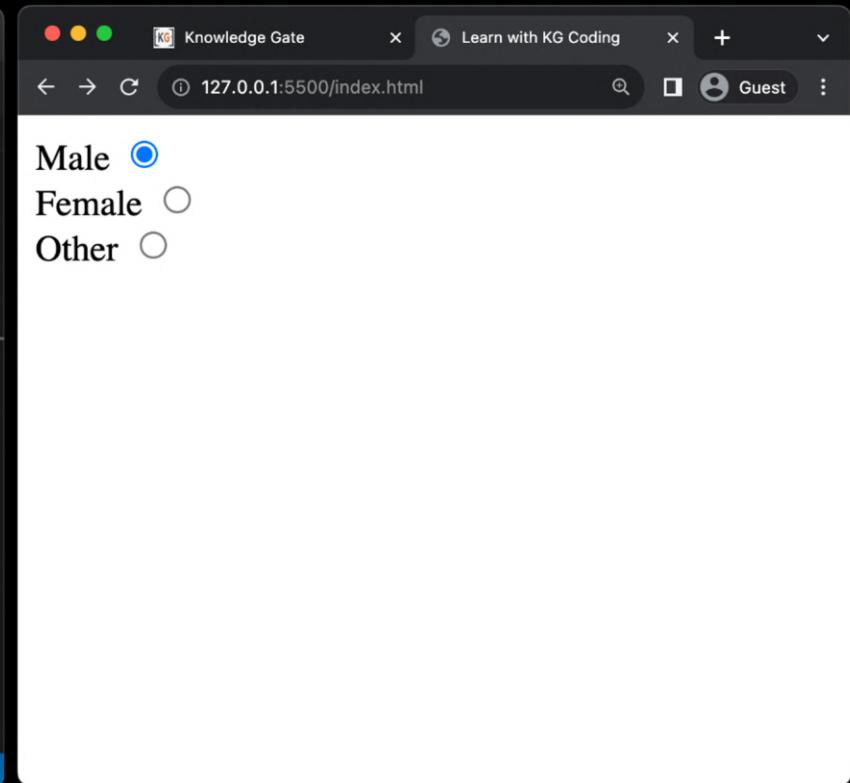
The screenshot shows a code editor with the file `index.html` open. The code defines a form with three radio buttons for gender selection:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <form>
        <label for="male">Male</label>
        <input type="radio" id="male" name="gender" value="Male"><br>

        <label for="female">Female</label>
        <input type="radio" id="female" name="gender" value="Female"><br>

        <label for="other">Other</label>
        <input type="radio" id="other" name="gender" value="Other"><br>
    </form>
</body>
</html>
```

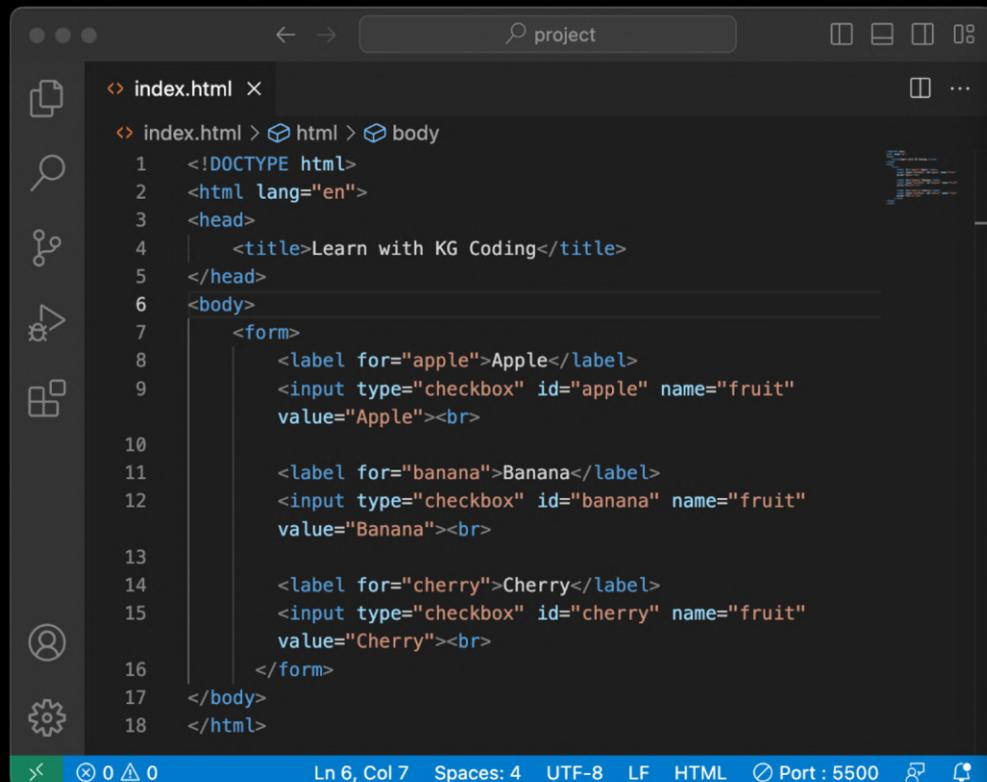
The code editor interface includes a sidebar with various icons for file operations, a search bar, and a status bar at the bottom indicating the current line and column.



The screenshot shows a web browser window titled "Knowledge Gate" with the URL `127.0.0.1:5500/index.html`. The page displays three radio buttons labeled "Male", "Female", and "Other". The "Male" option is selected, indicated by a blue outline around the radio button and the word "Male" in bold.

Male Male
Female Female
Other Other

3.5 Input type: Checkbox

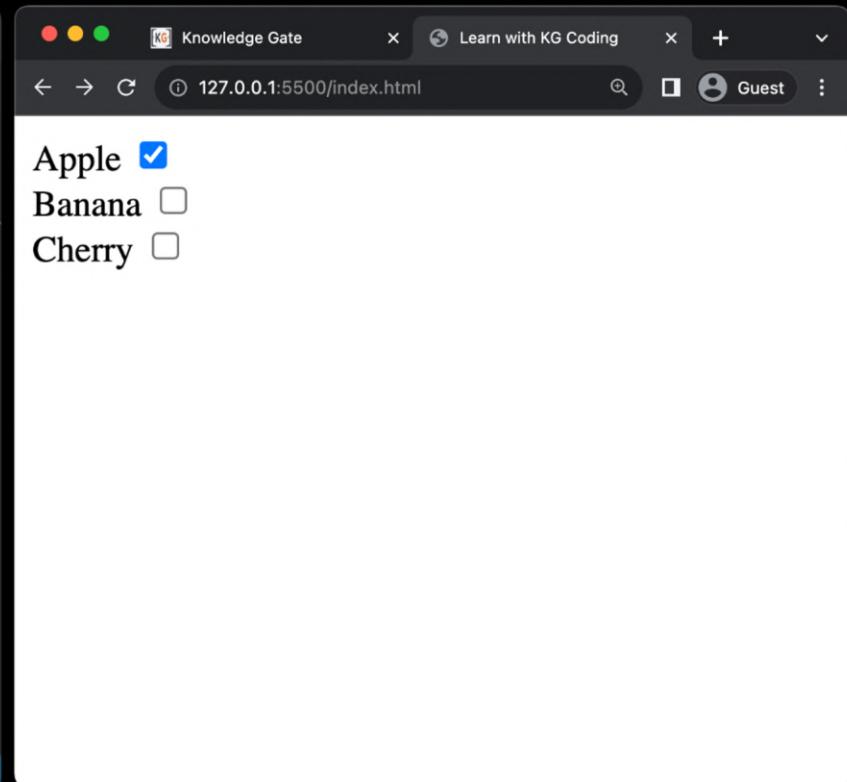


A screenshot of a code editor showing the file `index.html`. The code defines a form with three checkboxes labeled Apple, Banana, and Cherry. The first checkbox for Apple is checked.

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

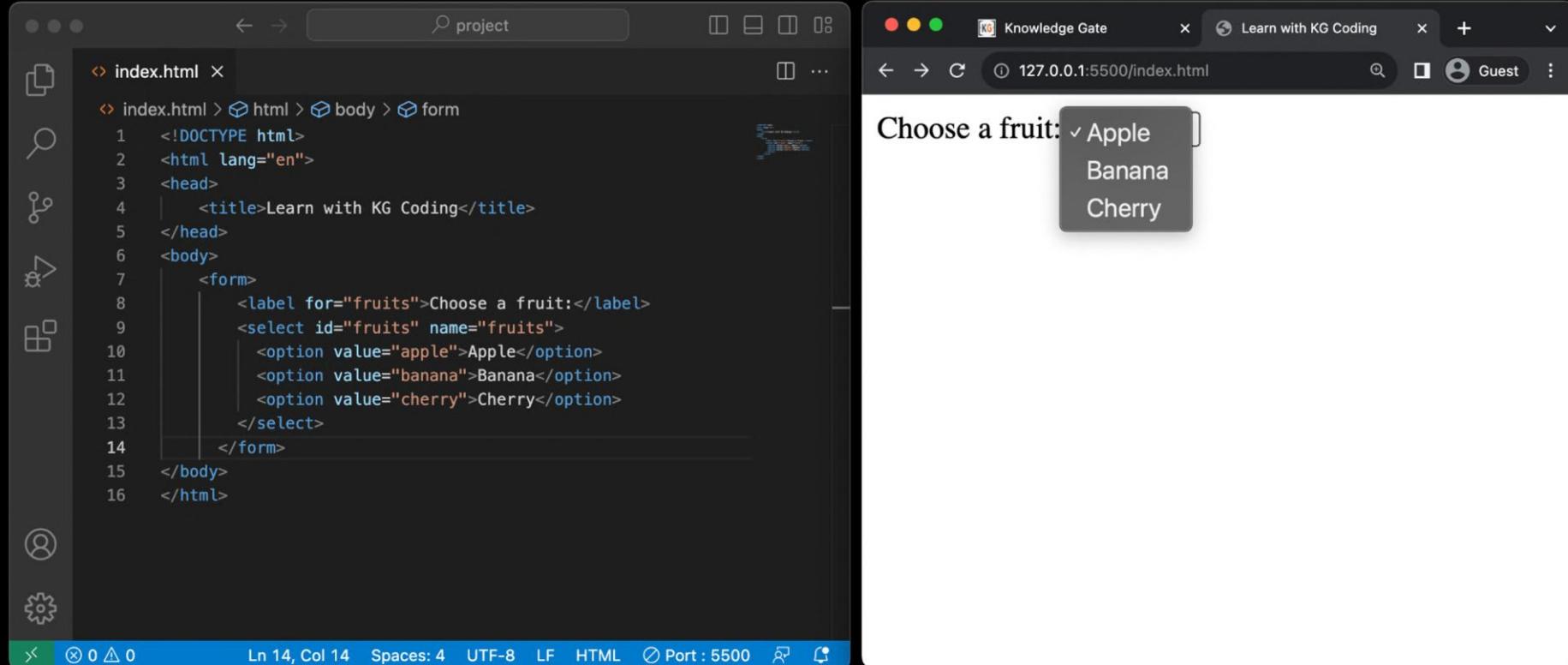
        <label for="cherry">Cherry</label>
        <input type="checkbox" id="cherry" name="fruit" value="Cherry"><br>
    </form>
</body>
</html>
```



A screenshot of a web browser displaying the rendered HTML from the code editor. The page shows three labels: "Apple" with a checked checkbox, "Banana" with an unchecked checkbox, and "Cherry" with an unchecked checkbox.

Apple
Banana
Cherry

3.5 Input type: Select

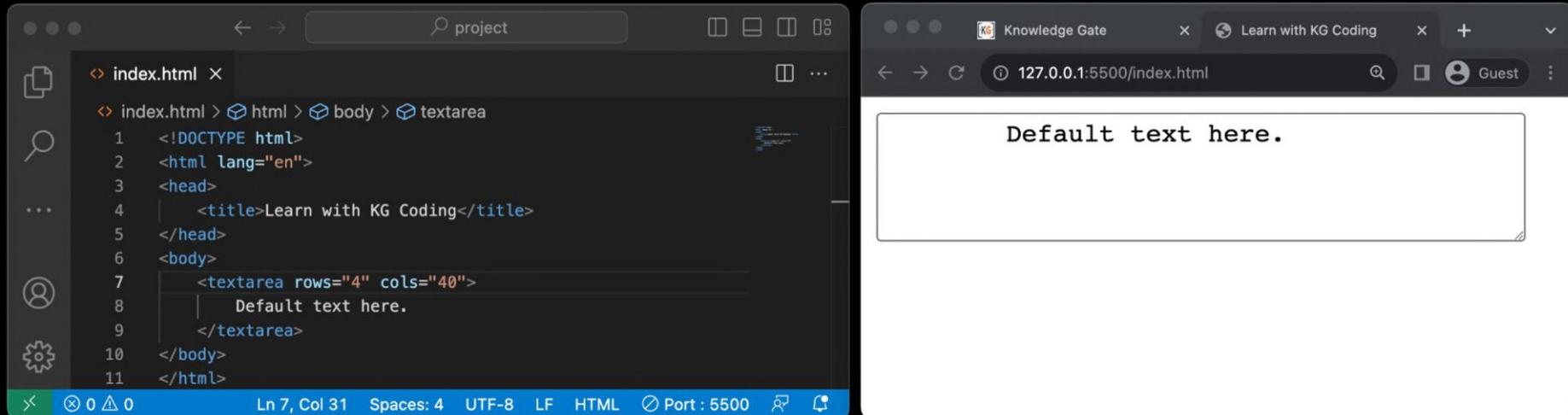


The image shows a code editor on the left and a web browser on the right. The code editor displays the file `index.html` with the following content:

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

The browser window on the right shows the rendered HTML. A modal dialog box is displayed with the text "Choose a fruit:" followed by a list of three options: "Apple", "Banana", and "Cherry". The option "Apple" is selected, indicated by a checked checkbox icon.

3.5 Input type: TextArea



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:

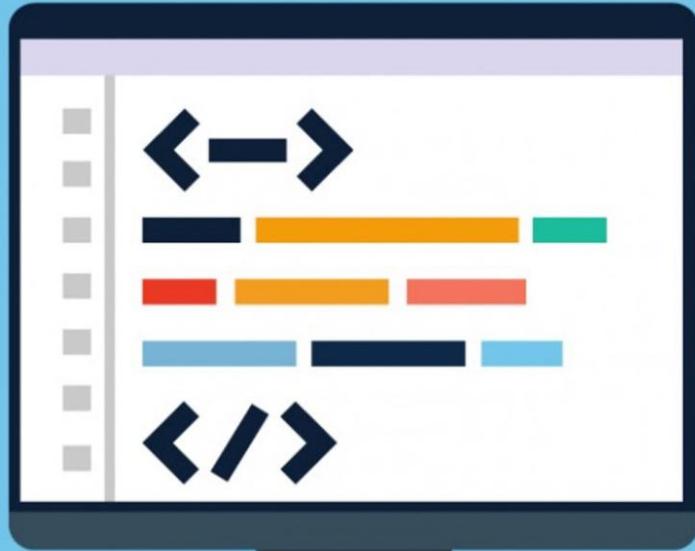
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Learn with KG Coding</title>
</head>
<body>
    <textarea rows="4" cols="40">
        Default text here.
    </textarea>
</body>
</html>
```

The browser window shows the rendered HTML at the URL 127.0.0.1:5500/index.html. It displays 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 textarea.

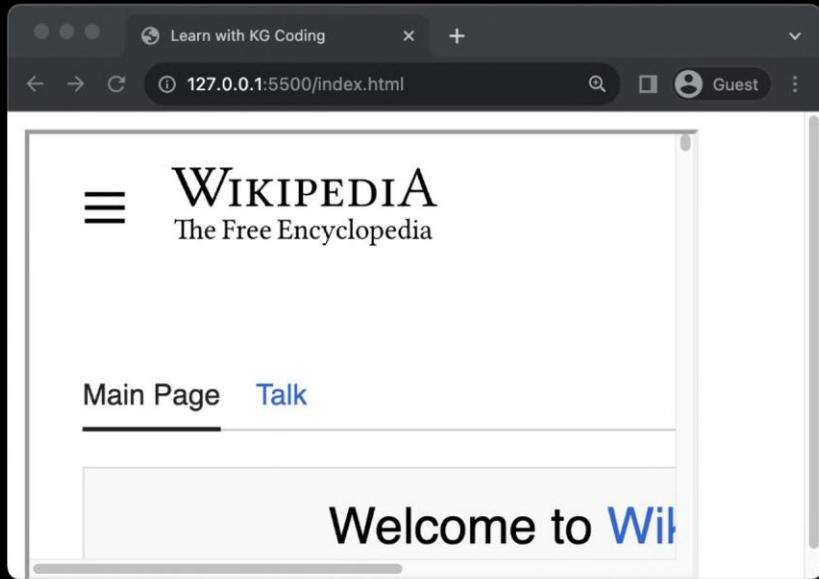
Level 5

List, Tables & Forms



4. iFrame Tag

4.1 Using iFrames



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

```
<!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>
```

The browser window shows the rendered output of this code. It features a header with three horizontal bars and the text 'WIKIPEDIA The Free Encyclopedia'. Below the header, there are two buttons: 'Main Page' (underlined) and 'Talk'. At the bottom of the page, the text 'Welcome to Wi' is visible.

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.

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

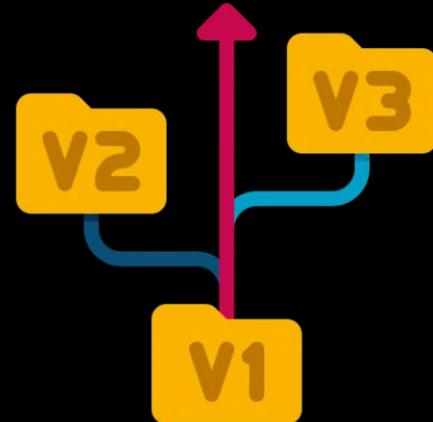
Github Pages & CodeSpace



1. GitHub

1.1 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.

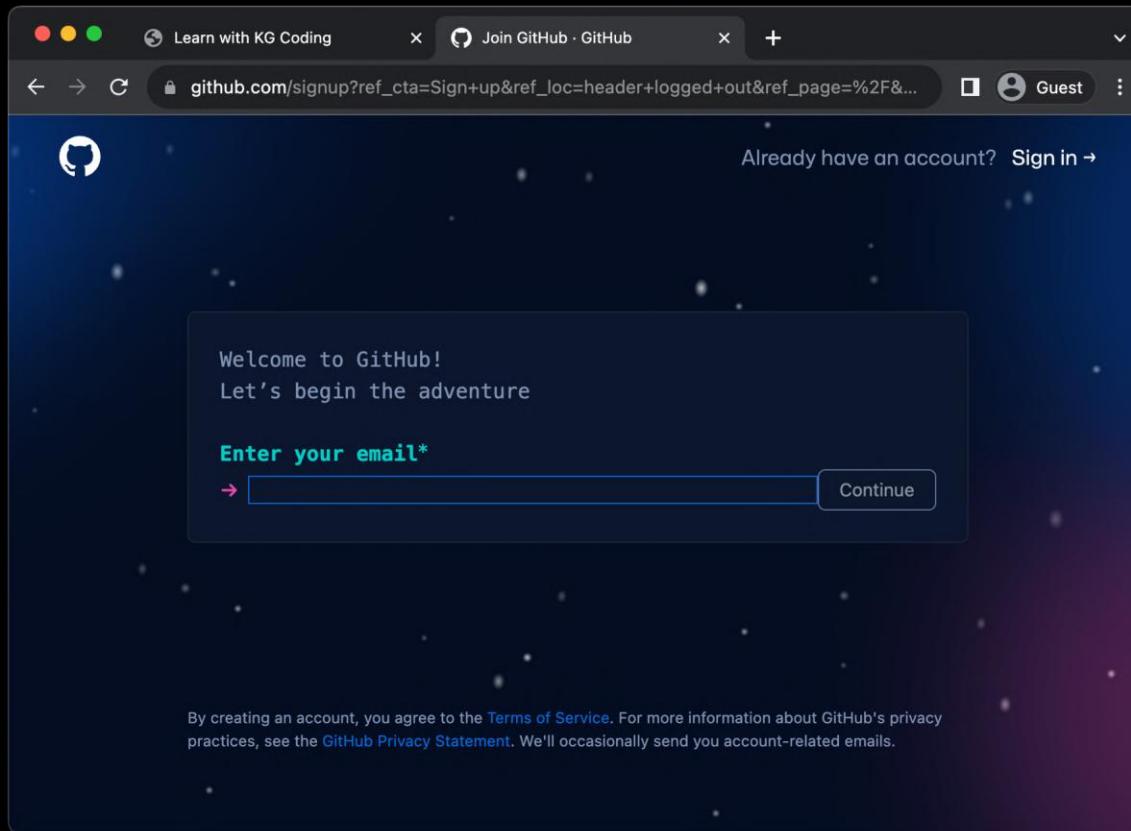


What is GitHub?

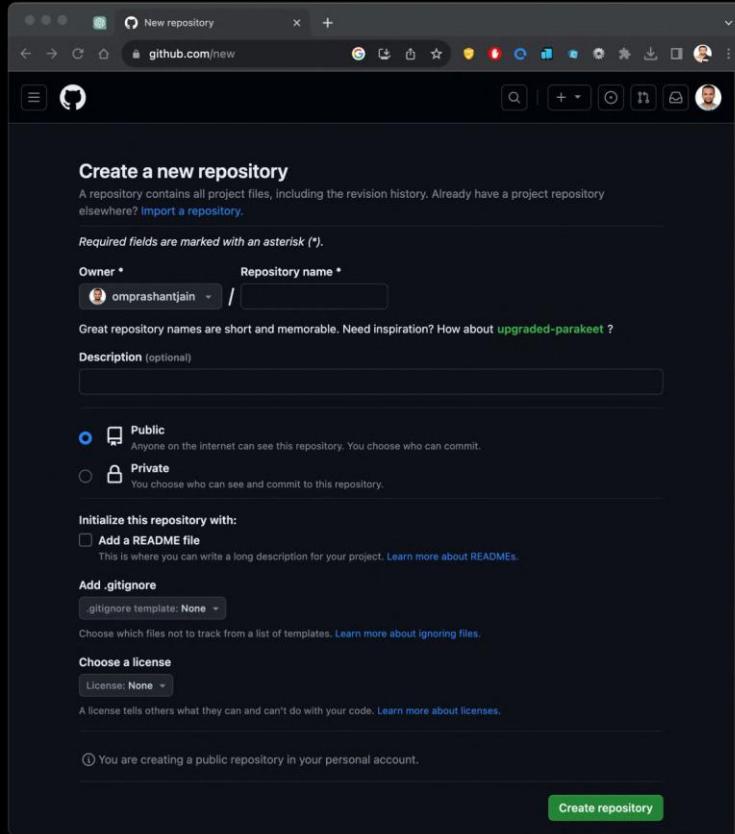
- **Definition:** A web service for hosting and collaborating on Git repositories.
- **Fork:** Creates a personal copy of another user's repository.
- **Pull Request:** A way to propose changes to existing code.
- **Issues:** Used for tracking bugs and feature ideas.



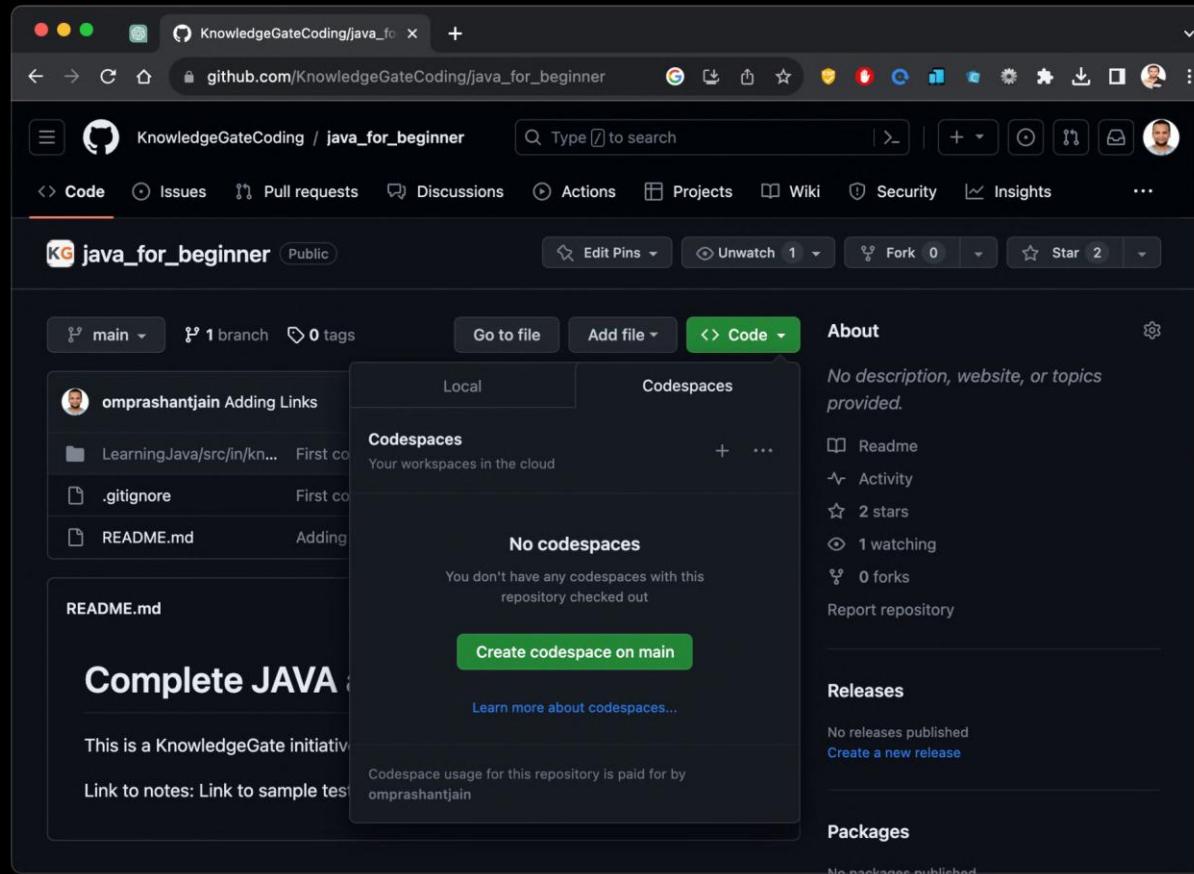
1.3 Account Creation



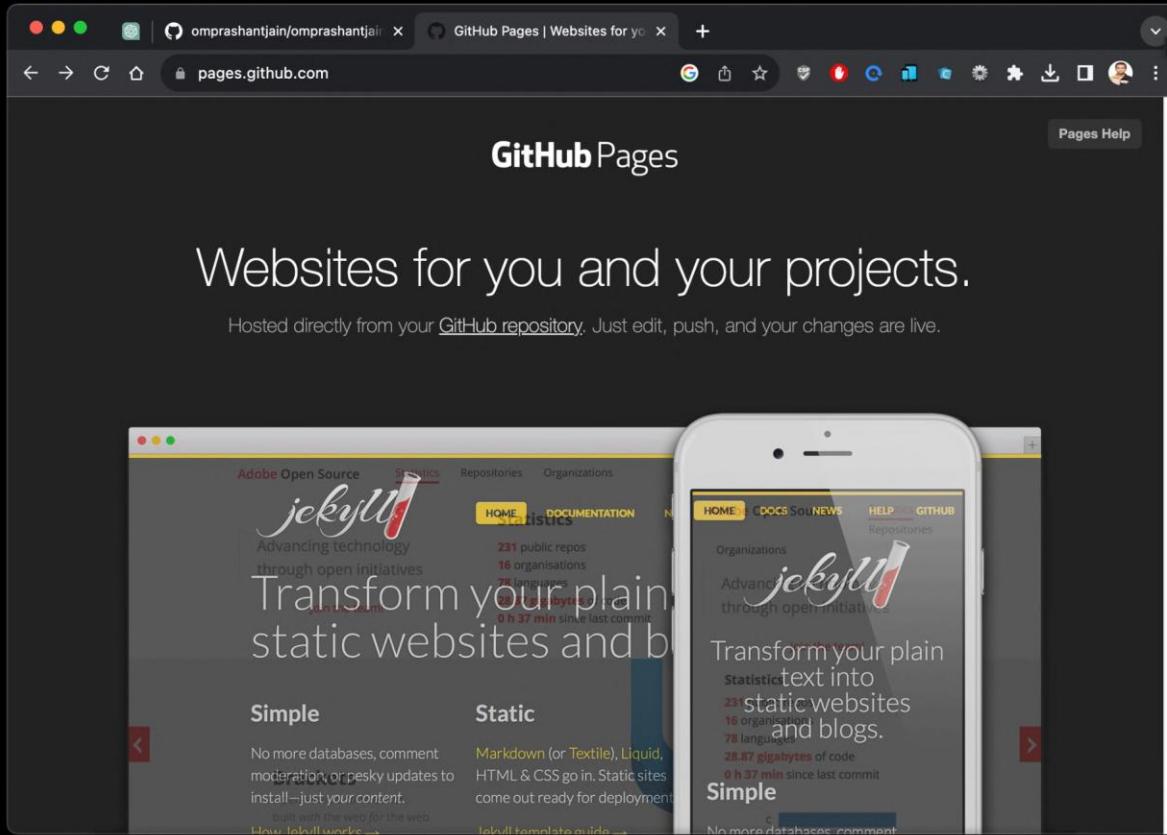
1.4 Creating a Repo



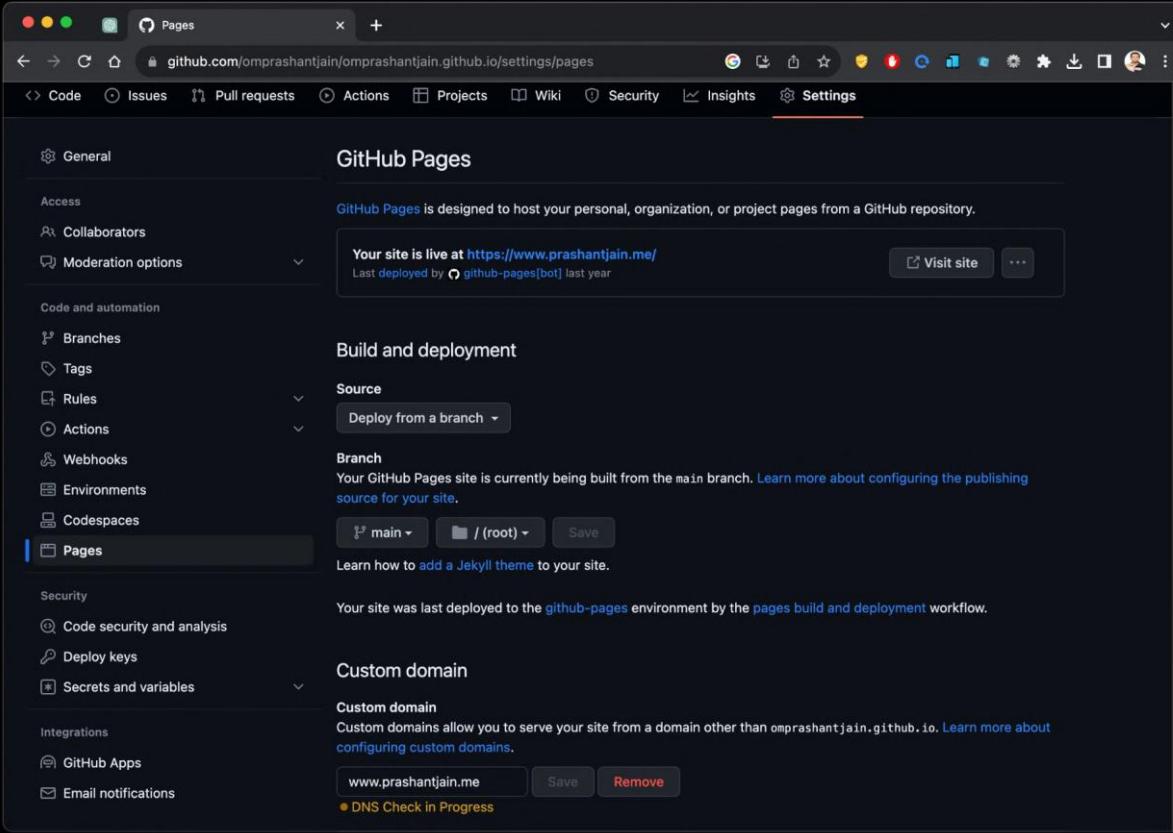
1.5 Creating a CodeSpace



1.6 Creating a Github Page



1.7 Publishing our Project



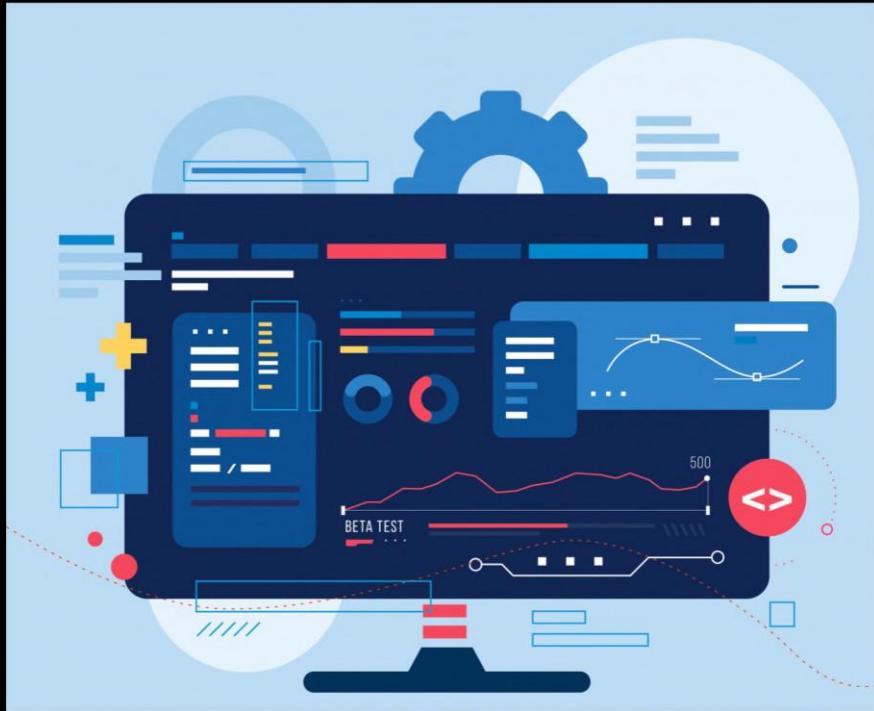
The screenshot shows the GitHub Pages settings page for a repository. The main header reads "GitHub Pages". On the left, there's a sidebar with various settings categories: General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages), Security (Code security and analysis, Deploy keys, Secrets and variables), Integrations (GitHub Apps, Email notifications), and Insights.

The "Pages" section in the sidebar is currently selected and highlighted in blue. The main content area displays the following information:

- GitHub Pages**: GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.
- Your site is live at <https://www.prashantjain.me>**. Last deployed by [github-pages\[bot\]](#) last year. Buttons for "Visit site" and "..." are present.
- Build and deployment**:
 - Source**: Deploy from a branch. Options include "main" and "/ (root)".
 - Branch**: Your GitHub Pages site is currently being built from the [main](#) branch. Learn more about configuring the publishing source for your site.
 - Learn how to add a Jekyll theme** to your site.
- Custom domain**:
 - Custom domain**: Custom domains allow you to serve your site from a domain other than [omprashantjain.github.io](#). Learn more about configuring custom domains.
 - A field shows the custom domain www.prashantjain.me, with a "Save" button and a red "Remove" button. A status message indicates a "DNS Check in Progress".

Level Bonus

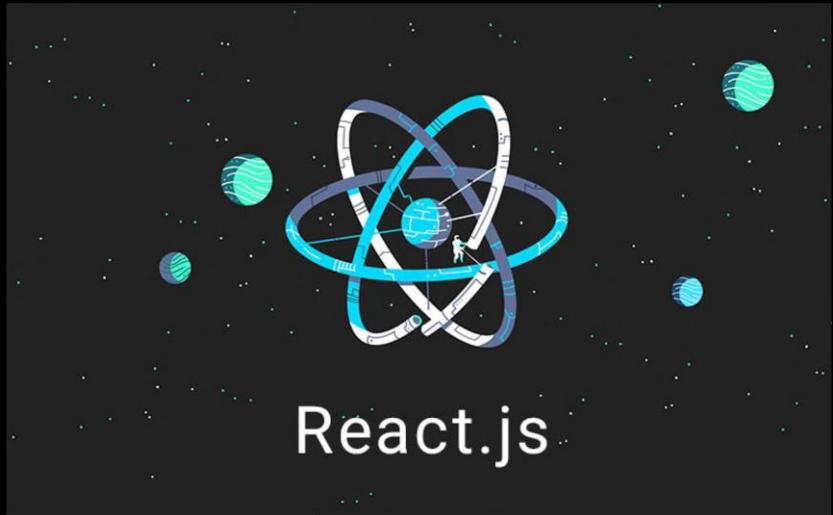
Github Pages & CodeSpace



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.



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.

