50+ HTML Interview

Question & Answer

Mallikarjun | @CodeBustler

HTML is very simple and **easy to learn** and In interview questions will be very less

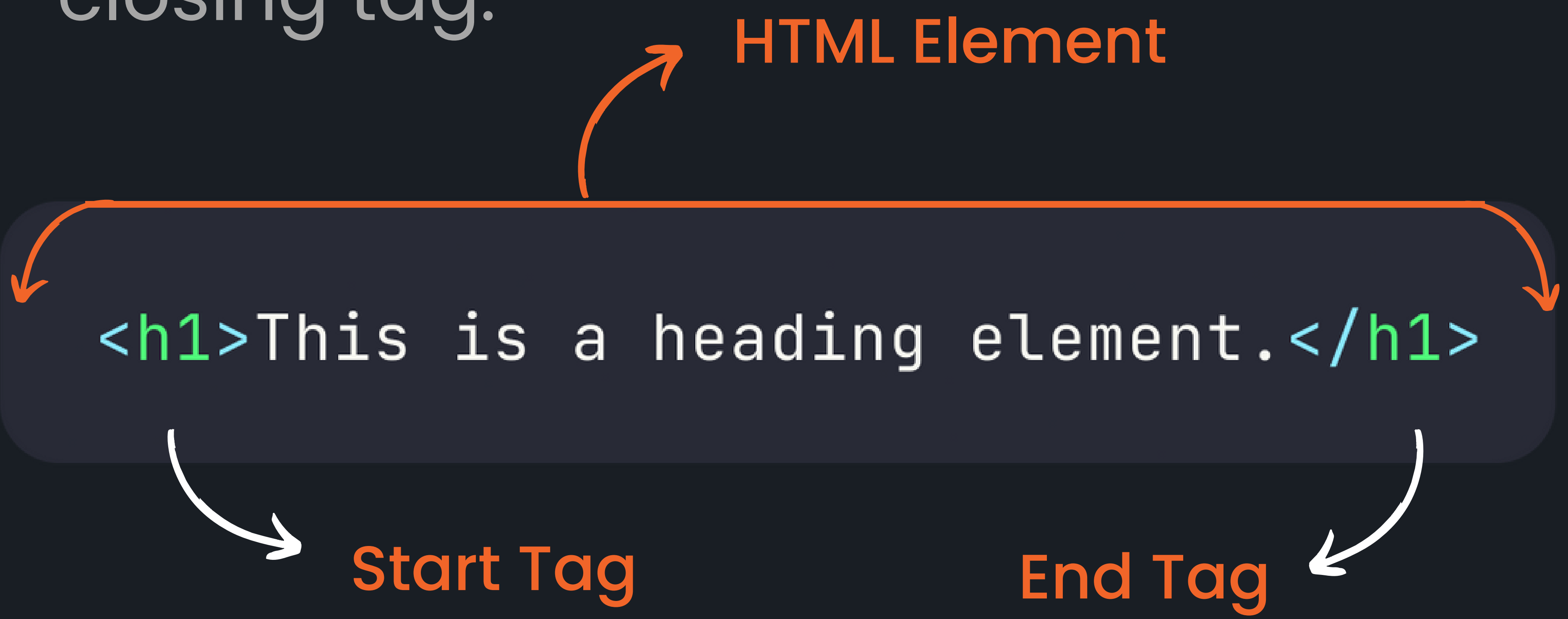
but you must know about this top **50+ important** question & answers

@CodeBustler

1. Diffn between HTML **Elements** & **Tags**

HTML Elements : are fundamental for webpage structure, including start and end tags plus content. They define content meaning and purpose, like <h1> for headings or <p> for paragraphs.

HTML Tags : are the actual codes used to define & create HTML elements. A tag consists of angle brackets (< and >) and usually encloses an element. Tags come in pairs: an opening tag and a closing tag.



```html
<h1>This is a heading element.</h1>
```

2. Should there be **multiple <h1>** tags?

No, there should be **only one <h1> tag** per webpage. It's **important for SEO** & document structure. Use <h2> - <h6> for subsections.

3. What are **new elements** in **HTML5** ?

1. <header>
2. <nav>
3. <section>
4. <article>
5. <aside>
6. <footer>
7. <main>
8. <figure>
9. <figcaption>
10. <video>
11. <audio>
12. <canvas>
13. <datalist>
14. <output>
15. <details>
16. <summary>
17. <mark>

@codebustler

4. What are the new **form elements** in **HTML5** ?

1. <input type="email">
2. <input type="url">
3. <input type="tel">
4. <input type="number">
5. <input type="range">
6. <input type="date">
7. <input type="time">
8. <input type="datetime-local">
9. <input type="month">
10. <input type="week">
11. <input type="color">
12. <input type="search">
13. <input type="file">
14. <textarea>
15. <keygen>
16. <output>

5. Difference between **semantic** & **syntactic** elements.

Semantic elements deal with the meaning of words and phrases, & convey ideas. **e.g**

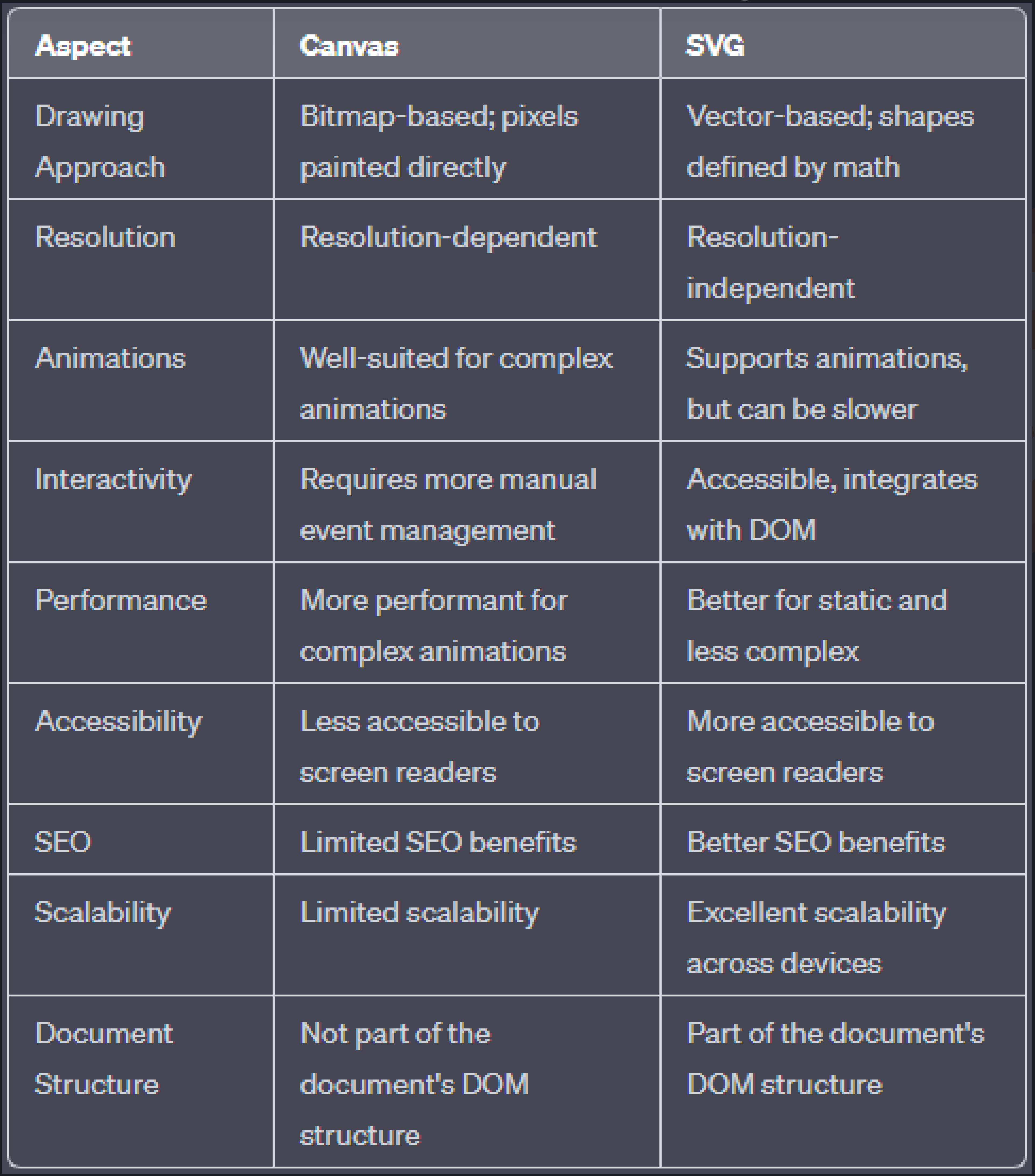
<header>	<section>	<address>
<nav>	<aside>	<time>
<main>	<footer>	<strong>
<article>	<figure>	<mark>

---

syntactic elements deal with the arrangement & structure of those words & phrases in a sentence/code, this ensure grammatical correctness & logical order. **e.g**

<html>	<h1> - <h6>	<form>
<head>	<ul>, <ol> & <li>	<input>
<body>	<table>, <tr>	<select>

6. Difference between **Canvas** & **SVG**



Aspect	Canvas	SVG
Drawing Approach	Bitmap-based; pixels painted directly	Vector-based; shapes defined by math
Resolution	Resolution-dependent	Resolution-independent
Animations	Well-suited for complex animations	Supports animations, but can be slower
Interactivity	Requires more manual event management	Accessible, integrates with DOM
Performance	More performant for complex animations	Better for static and less complex
Accessibility	Less accessible to screen readers	More accessible to screen readers
SEO	Limited SEO benefits	Better SEO benefits
Scalability	Limited scalability	Excellent scalability across devices
Document Structure	Not part of the document's DOM structure	Part of the document's DOM structure

7. Difference between **class** & **attributes**

Class :

- Used for styling & grouping elements.
- Applied to one or more elements.
- Defined with **class** attribute in HTML.
- Referenced in CSS for styling rules.
- Enables consistent styling across elements.

Attributes :

- Provide inform/settings to elements.
- Offer instructions for behavior/display.
- Elements can have multiple attributes.
- Examples: **src**, **alt**, **href**, **class**.
- Enhance functionality and appearance.

@CodeBustler

8. How is **DOM rendered** by Browser ?

- The browser reads HTML,
- creates a structured tree (DOM),
- applies styles,
- figures out where things go,
- paints & layers them, and
- displays the final page on screen.

9. Different **ways of adding Styles** HTML

- **Inline Styles** : Add styles directly within HTML tags using the style attribute.
- **Internal Stylesheet** : Place styles within a <style> tag in the HTML <head>. Apply to elements using selectors.
- **External Stylesheet**: Link an external CSS file to the HTML using <link>. reusable across multiple pages.

10. Difference between **Async** & **Defer**

Both the **async** and **defer** attributes are used in HTML to **control the execution of external scripts**, but they have different effects on how the scripts are loaded and executed in relation to the HTML document

- Use **async** when the script is independent of the DOM and can run as soon as it's downloaded.

- Use **defer** when the script needs to wait for the DOM to be fully parsed and constructed before execution.

@CodeBustler

11. What is **responsive** web design

Responsive web design (RWD) is an approach to web design & development that aims to create websites that automatically **adapt** and **provide** an optimal viewing experience on various **devices & screen sizes**, including desktops, laptops, tablets, and smartphones.

Key principles of responsive web design

- Fluid Grids
- Flexible Images
- Media Queries
- Viewport Meta Tag
- Content Prioritization
- Breakpoints

@CodeBustler

12. What does **<!DOCTYPE html>** do ?

The <!DOCTYPE html> declaration specifies the version of HTML (HTML5) the web page follows.

It helps browsers display the page correctly and consistently across different devices and avoids compatibility issues

13. What is a **self closing** tag ?

A self-closing tag is like a quick way to add something to a web page without needing to open and close it separately.
e.g
<img>, <br>, <hr>, <link>
<input>, <meta> & <col> etc...

@CodeBustler

14. Can a page have multiple **<header>** ?

Yes, you can have multiple <header> elements in an HTML document. The <header> element is used to represent introductory content at the beginning of a **section** or a **container**.

It is often used within structural elements like **<article>**, **<section>**, or the **<body>** itself.

Example

```
<section>
  <header>
    <h2>About Us</h2>
    <p>Lorem ipsum dolor sit, amet, consectetur</p>
  </header>
  <p>Content about the company goes here...</p>
</section>
```

@CodeBustler

15. What is the use of **META** tags ?

A meta tag is an HTML element placed in the <head> section of a web page.

It provides extra information about the page for browsers & search engines, like

- character encoding
- viewport settings
- page description
- author, and more.

Meta tags enhance how the page is displayed, indexed, and understood by different technologies.

```html
<meta charset="UTF-8" />
```

Example

16. Diffn between **attribute** & **property**

Attributes are defined in HTML and provide initial values to elements.

```
<img src="image.jpg" alt="Example Image">
```

In this example, the **src** & **alt** attributes are used to provide **initial values** to an **<img>** element.

Properties are part of the DOM & represent the current state of elements, accessible and manipulable using JS.

```
inputElement.value // Output: Initial Value
```

In this example, the value property is JS to interact with an <input> initial value.

17. What is an **iframe**? How it works ?

An <iframe> (short for inline frame) is an HTML element that allows you to embed another HTML document within the current document.

It essentially creates a "window" within your web page that displays content from another source, such as a different website or a different part of the same website.

It's like a "mini browser window" embedded within your page.

```html
<iframe src="https://www.example.com"></iframe>
```

Example

How iframe **works**

- **Embedding**: Use <iframe> to put a mini web page inside your page.
- **Display**: It's like a little window showing another website or video.
- **Interactivity**: People can use it like a small part of the page.
- **Security**: It has some rules to prevent problems.
- **Attributes**: <iframe> has settings like size and source. (src, width, height, etc.)

@CodeBustler

iframe common use cases

Common use cases for iframes include

- Embedding videos
- Displaying maps
- Showing social media content, and
- Integrating content from other websites within your own.

18. What is the use of the **"alt"** attribute in ?

The "alt" attribute in an <img> tag is used to provide alternative text for the image.

This text is displayed if the image cannot be loaded or if the user is using assistive technologies like screen readers.

19. What is the **Geolocation API** ?

The Geolocation API is a web technology that allows websites to **access and retrieve a user's geographical location** information, such as latitude and longitude, using JavaScript.

This enables websites to **provide location-aware services** and customize content based on the user's location. It can be used for various purposes,

- including mapping,
- location-based search,
- navigation, and more.

Users are typically **prompted to give permission** before their location data is accessed.

20. What are the **web workers** ?

Web Workers are a JavaScript feature that enables running background tasks independently of the main user interface (UI) thread.

They allow for multitasking, handling **heavy computations**, and maintaining a responsive UI.

Web Workers communicate with the main thread using messages and are used for tasks like calculations, data processing, and real-time updates without blocking the UI.

@CodeBustler

21. Diffrn between **progress** & **meter** tag

<progress> : Shows task completion progress, often for uploads, with dynamic updates using JS. Uses attributes like **value** and **max** for the current and maximum values. Displays as a filling horizontal bar.

```
<progress value="70" max="100">70%</progress>
```

<meter>: Displays measurements within a range (e.g., ratings, temperatures) with built-in min/max values. Attributes like **value**, **min**, & **max** set values. Shows as a bar with a pointer, often verti/horizont.

```
<meter value="8" min="0" max="10">8 out of 10</meter>
```

22. Which **Video/Audio formats** supported by HTML

HTML supports diverse video/audio formats, but **compatibility varies.** To ensure wider playback, use multiple formats

Video Formats:

1. MP4 (H.264)
2. WebM (VP8/VP9)
3. Ogg (Theora)

Audio Formats:

1. MP3
2. AAC
3. Ogg (Vorbis)

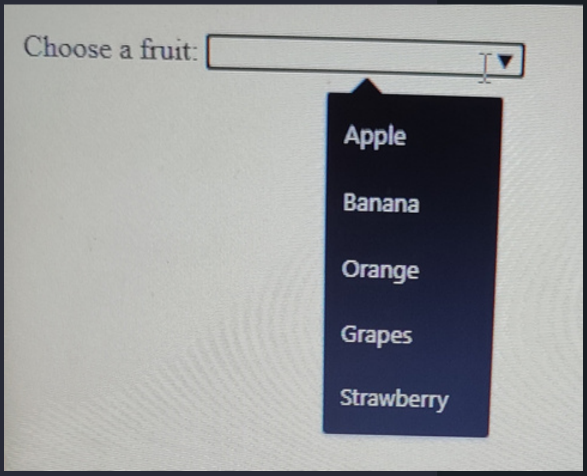
@CodeBustler

23. What is **<datalist>** tag | example

The <**datalist**> tag offers pre-defined options for an <**input**>, enabling **autocomplete**. It links with the input using **matching id** & **list attributes**, showing a dropdown of suggestions as users type

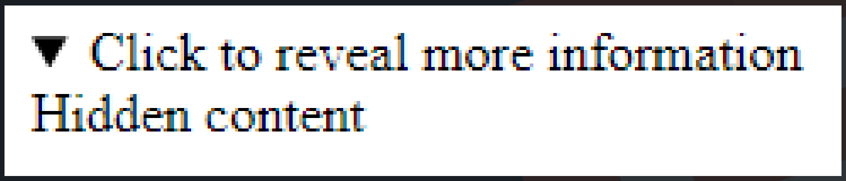
@CodeBustler

```
<label for="fruits">Choose a fruit: </label>
<input
  type="text"
  id="fruits"
  list="fruit-options"
/>

<datalist id="fruit-options">
  <option value="Apple"></option>
  <option value="Banana"></option>
  <option value="Orange"></option>
  <option value="Grapes"></option>
  <option value="Strawberry"></option>
</datalist>
```



24. Use of <**details**> & <**summary**>

The <**details**> & <**summary**> elements in HTML are used to create interactive disclosure widgets that allow you to **hide or reveal content**.



▼ Click to reveal more information
Hidden content

<details> Element :

- Acts as a container to toggle content.
- Creates an expandable/collapsible widget.
- Content inside <details> starts hidden.

<summary> Element :

- Offers a visible label/heading for widget.
- When collapsed, only <summary> content shows.

@CodeBustler

25. Explain about **<a>** anchor tag & **target** attribute

Creates clickable links to other content. Uses the **href** attribute to specify the link's destination

target Attribute:

Used within **<a>** to how displayed.

- **_blank**: Opens in a new window/tab.
- **_self**: same window/tab (default).
- **_parent**: parent frame (iframes).
- **_top**: full window (breaks frames).

```html
<a href="https://www.google.com"
  target="_blank"> Google
</a>
```

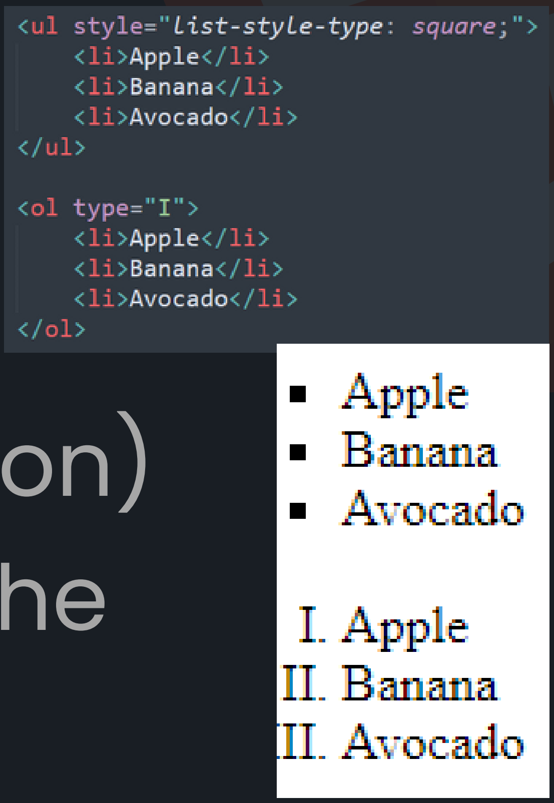
@CodeBustler

26. Explain in detail about **lists** in HTML

Lists in HTML are used to **organize** & **present information** in an ordered or unordered format.

There are three main types of lists

1. Ordered List **<ol>**
2. Unordered List **<ul>**
3. Description List **<dl>** Uses **<dt>** (term) & **<dd>** (definition) elements explanation of the corresponding term



```html
<ul style="list-style-type: square;">
    <li>Apple</li>
    <li>Banana</li>
    <li>Avocado</li>
</ul>

<ol type="I">
    <li>Apple</li>
    <li>Banana</li>
    <li>Avocado</li>
</ol>
```

```
▪ Apple
▪ Banana
▪ Avocado

  I. Apple
 II. Banana
III. Avocado
```

The **type** attribute in HTML is used to specify the type of marker/numbering for list items in **<ol>.** eg : letters, Numbers Roman Num | for <ul> using css styling

@CodeBustler

27. How do you serve a page with content in **multiple languages** ?

Serve multi-language web page:

1. Set **lang** attrbute in <html> for default.
2. Use **lang** for content elements.
3. Add <meta charset="UTF-8">
4. Optionally, link alternate versions, Using <link> tag
5. Consider fonts, switchers, and tools.
6. Ensure dynamic content localization.

This ensures a user-friendly experience across languages.

```html
<html lang="en">
```

default language of the entire page

28. What are **data-** attributes ?

Data attributes (also known as **"data-*"** attributes) in HTML are custom attributes that allow you to **store extra information** or data directly within HTML elements.

They are prefixed with **"data-"** followed by a descriptive name.

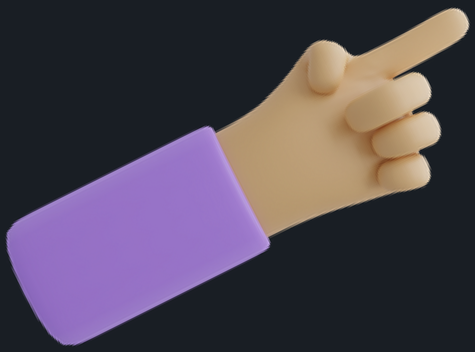
```
<div  id="myElement" data-custom-value="42">
Custom element </div>

<script>
  var element = document.getElementById("myElement");
  var customValue = element.getAttribute("data-custom-value");
  console.log(customValue); // Outputs: 42
</script>
```

@CodeBustler

29. Why is it generally a good idea to position CSS <**link**>s between <head></head> and <script>s just before </body>?

This is generally a **good practice** to position CSS <link> elements between the <head> and JS <script> elements just before the closing </body> tag,

This practice enhances user experience and page loading efficiency.



@CodeBustler

Advantages :

- **Faster Load** : CSS loads early, reducing "flash of unstyled content".
- **Progressive Rendering** : Page starts rendering sooner, improving user perception.
- **Parsing Efficiency**: HTML parsing isn't blocked by CSS loading.
- **Caching**: CSS files are cached for subsequent visits.
- **Consistency**: Follows standard HTML structure for clarity.

Exceptions :

- **Critical CSS** : Inline critical styles for immediate rendering.
- **Async/Defer Scripts**: Order matters less with async/defer attributes.
- **Advanced Optimizations** : Specific optimizations might impact placement.
- **Conditional Loading**: Load CSS/scripts conditionally based on circumstances.

@CodeBustler

30. What is **progressive rendering**?

Progressive rendering ensures quick and engaging page loads:

- Focuses on delivering content smoothly and fast.
- Displays crucial content first for user interaction.
- Delays non-essential resources like images.
- Boosts user engagement and satisfaction.
- Often linked with responsive design.
- Uses caching, async loading, and deferred execution.
- Measured by metrics like First Contentful Paint and Time to Interactive.

31. Why would you use a **srcset** attribute in an image tag? **Explain the process** the browser uses when evaluating the content of this attribute

@CodeBustler

The **srcset** attribute specifies the URL of the image to use in different situations.

This attribute is required when <**source**> is used in <**picture**>.

```html
<picture>
  <source media="(min-width:650px)" srcset="img1.jpg"/>
  <source media="(min-width:465px)" srcset="img2.jpg"/>
  <img src="img3.jpg" alt="Flowers" style="width:auto;"/>
</picture>
```

In this example 3 different images will be displayed in 3 different width of screen.

32. What are **empty elements** in HTML ?

Empty elements in HTML are elements that don't have closing tags because they don't contain content.

Also known as **self-closing** or **void elements**

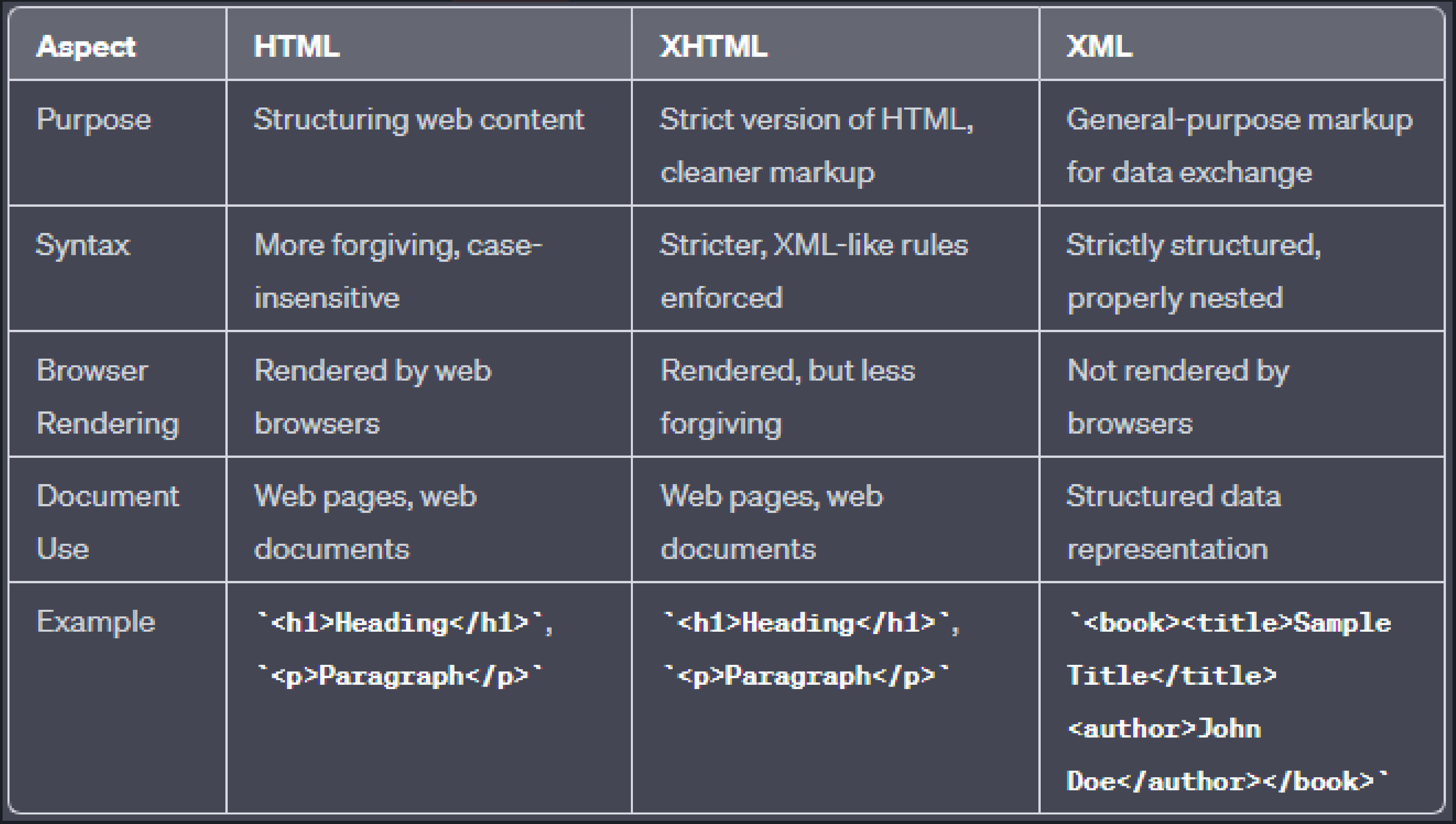
They are self-contained and often used for specific purposes, like

- images (<img />)
- line breaks (<br/>)
- Meta (<meta/>)
- horizontal rule (<hr/>) and
- input fields (<input/>)

@CodeBustler

33. HTML **Vs** XHTML **Vs** XML

@CodeBustler



Aspect	**HTML**	**XHTML**	**XML**
Purpose	Structuring web content	Strict version of HTML, cleaner markup	General-purpose markup for data exchange
Syntax	More forgiving, case-insensitive	Stricter, XML-like rules enforced	Strictly structured, properly nested
Browser Rendering	Rendered by web browsers	Rendered, but less forgiving	Not rendered by browsers
Document Use	Web pages, web documents	Web pages, web documents	Structured data representation
Example	`<h1>Heading</h1>`, `<p>Paragraph</p>`	`<h1>Heading</h1>`, `<p>Paragraph</p>`	`<book><title>Sample Title</title> <author>John Doe</author></book>`

33. What are the various **formatting tags**

Formatting tags in HTML are used to apply **visual styling and structure to text** content within a web page.

1. <b>
2. <i>
3. <u>
4. <strong>
5. <em>
6. <mark>
7. <sub>
8. <sup>
9. <del>
10. <ins>
11. <code>
12. <pre>
13. <small>
14. <s>
15. <kbd>
16. <var>
17. <samp>
18. <time>
19. <dfn>

@CodeBustler

34. Significance of <**head**> & <**body**> tags

<head> Tag :

The **<head>** tag holds non-displayed metadata like **title**, **encoding**, **styles**, and **scripts**. It guides search engines and browsers on how to handle and present the page. (SEO)

<body> Tag :

The <**body**> tag surrounds visible content users interact with, like **headings**, **images**, and **links**. It defines the main content, **appearance**, and **interactivity** of the webpage.

35. Explain **Inline** & **block** elements

Inline Elements :

- Flow within text or content.
- Take up only as much width as needed.
- Examples:
  <span>, <a>, <strong>, <em>, <img>.

Block Elements :

- Create new blocks &start on new lines.
- Occupy full available width by default.
- Examples:
  <div>, <p>, <h1>, <ul>, <li>.

36. What is the use of **span** tags ?

Used to apply **specific styles**, target individual words/phrases for styling or scripting, & enhance accessibility within a larger block of text on a webpage

37. List the media types and formats supported by HTML

@CodeBustler

Images:

- Formats: JPEG, PNG, GIF, SVG
- Elements: <img>, <picture>

Audio:

- Formats: MP3, WAV, Ogg
- Elements: <audio>

Video:

- Formats: MP4, WebM, Ogg
- Elements: <video>

Embedded Content:

- Formats: PDF, Flash (SWF)
- Elements: **<object>**, **<embed>**, **<iframe>**

Fonts:

- Formats: TTF, OTF, WOFF, WOFF2
- Elements: **<link>**, **@font-face** (CSS)

Interactive Content:

- Formats: JavaScript, WebGL
- Elements: **<script>**

Stylesheets:

- **Formats: CSS**
- **Elements: <link>, <style>**

Data:

- **Formats: JSON, XML**
- **Elements: None (accessed via JavaScript)**

38. How can we include Google maps on a website?

To embed a Google Map into an HTML page, you can use the 2 methods.

- Google Maps Embed **API** or
- Use an <**iframe**> tag with **map's URL**.

1. Google Maps Embed API :

Get an API Key:

- Go to the Google Cloud Console.
- Create a **new project** or select an existing project.
- Enable the "**Maps JavaScript API**" for your project.
- Create an **API Key** by navigating to "APIs & Services" > "Credentials".

2. Use an <iframe> tag with map's URL:

1. **Locate the Map on Google Maps:**
   - Go to Google Maps.
   - Find the location you want to embed on the map.
2. **Get the Embed Code:**
   - Click on the "**Share**" button
   - Choose the "**Embed a map**" tab.
   - Customize the map's size if needed.
   - Copy the generated **<iframe>** code.
3. **Paste the <iframe> Code:**
   - In your HTML file, paste the **<iframe>** code where you want the map to appear.

Google Map Embed Example :

```html
<iframe
  width="600"
  height="450"
  frameborder="0" style="border:0"
  src="URL_TO_EMBEDDED_MAP / URL_YOUR_API_KEY"
  allowfullscreen>
</iframe>
```

Embed with Gmap URL

Embed with API

@CodeBustler

39. What is HTML5? What are its new features that were not present in HTML?

- **Semantic elements** (e.g., <header>, <nav>, <article>) improved content structure.
- Native **audio/video** support via <audio> and <video> elements, no need for plugins.
- **<canvas>** enabled dynamic graphics, games, and animations with JavaScript.
- **Web workers** allowed background tasks, boosting performance.
- **localStorage** API stored data locally for offline web apps.
- **WebSockets** facilitated real-time browser-server communication.

- **Geolocation API** provided user location info for location-based services.
- Native **drag-and-drop** support for interactions in web apps.
- Enhanced forms: **new input types,** validation attributes, <datalist> for autocomplete.
- Supported **responsive design** with flexible layout and media queries.

40. Explain the **layout of HTML**.

1. **<!DOCTYPE> :** Specifies HTML version.
2. **<html> :** Root element.
3. **<head> :** Contains metadata (title, links, scripts).
4. **<body> :** Displays visible content (text, images, interactive elements).
   - Headings (<h1> to <h6>) & para <p>.
   - Images (<img>).
   - Hyperlinks (<a>).
   - Lists (<ul> and <ol>).
   - Tables (<table>).
   - Forms (<form> and input fields).
   - Interactive elements (buttons, multimedia).
   - Layout divisions (<div>, semantic elements like <header>, <main>, <footer>).

40. Explain an **image map** in HTML

An image map in HTML allows you to create clickable areas within an image. Here's how it works:

1. **Image Element (<img>):** Place the base image using the <img> tag.
2. **Map Element (<map>):** Use the <map> tag to define the map and hotspots.
3. **Hotspot Areas (<area>):** Define clickable areas with <area> tags, specifying **shape**, **coordinates**, and links.

```html
<img src="image.jpg" usemap="#imagemap">
<map name="imagemap">
    <area shape="rect" coords="0,0,100,100" href="link1.html">
    <!-- More <area> elements for other hotspots -->
</map>
```

When users click these hotspots, they're directed to specific URLs or actions.

41. How to add a **favicon** in HTML?

- Create a square image (e.g., 16x16 pixels).

- Upload the image to your website's directory.

- Inside the <head> section of your HTML, add:

  Replace "favicon.png" with your image's path.

```html
<link rel="icon" type="image/png" href="favicon.png">
```

- Browsers will display the favicon in tabs and bookmarks.

42. How do you create links to sections within the same page?

Use the <a> tag in HTML to create links, like <a href="**#topmost**"> BACK TO TOP</a>, which links to a bookmark named "topmost" defined with <a name="topmost"> on the same page.

43. What is the use of the figure tag in HTML 5?

The <figure> tag in HTML5 is used to group media content with captions. It provides semantic meaning to the relationship between the content and its caption.

```html
<figure>
    <img src="image.jpg" alt="Description">
    <figcaption>Caption goes here.</figcaption>
</figure>
```

44. How to comment single & multi line in HTML?

HTML single-line/multi line comment is
<!-- comment content -->

45. How to embed YouTube video in HTML?

1. On a computer, go to the **YouTube video** you want to **embed**.
2. Under the **video**, click SHARE .
3. Click **Embed**.
4. From the box that appears, copy the **HTML** code.
5. Paste the code into your blog or website **HTML**.

@CodeBustler

46. Which tags are no longer valid in HTML5?

<acronym>, <big>, <center>, <applet>, <frame>, <basefont>, <font>, <dir>, <isindex>, <frameset>, <noframes>, <tt>, <u>, <strike>, <s>

47. Which HTML5 tag would you use to define footer?

<footer> tag is used to define footer in HTML5.

Footer usually contains information like **authorship**, **copyright**, **back to top**, **contact**, **sitemap**, etc.

48. Which tag represents an independent piece of content of a document in HTML5?

The <**article**> tag in HTML5 represents standalone content like blog posts, forum entries, or news stories, meant to be distributed independently from the rest of the site.

49. How do you insert a copyright symbol on a browser page?

To insert the copyright symbol, you need to type **&copy;** or & **#169;** in an HTML file.

@CodeBustler

50. Do all HTML tags come in a pair?

No, there are single HTML tags that do not need a closing tag. Examples are the **<img>** tag and **<br>** tags.

@CodeBustler

Mallikarjun | Web Developer

Follow For More !

And i need dopamine ⚡