

HTML

Interview Questions

(Practice Project)



Interview Questions

Easy

Q1.What are tags and attributes in HTML? Explain a few commonly used tags and attributes.

Answer -

Tags are browser instructions that surround a keyword in angled brackets (< and >). They appear in pairs: an opening tag and a closing tag (</tag>). You wish to show the text that occurs between these tags. For example, anything between <p> and </p> will be displayed as a paragraph as the <p> element indicates a paragraph.

Additional details regarding how to handle that element may be found in the attributes, which are located within the opening tag. Think of these as comprehensive tagging instructions. An equal sign (=) and quotations surround the name and value that make up an attribute. Typical characteristics include the following:

- **href:** indicates the link's destination URL when used in combination with the tag (for anchors). Here's an illustration: <a href="<https://www.example.com>">Check out our website The user would be directed to the example website by clicking on the "Visit our website" link.
- **src:** specifies the path to the image file when used with the tag (for images). As an illustration, the HTML code would show the file "banner.jpg" along with the accessible alternative text "Logo."
- **alt:** In reference to accessibility, it is important to describe non-text information, such as photos, using the alt property. The purpose of the image is explained to visually challenged people through the use of screen readers.
- **id and class:** these attributes help assign unique identifiers or categorize elements. They are useful for styling with CSS (Cascading Style Sheets) to define a page's look and feel.

Q2.What is semantic HTML?

Answer -

Semantic HTML refers to using HTML elements that convey the meaning, or semantics, of the content they contain. Instead of focusing just on appearance, semantic HTML clarifies the content's purpose and structure for both browsers and developers.

Think of it like this: non-semantic HTML elements are like plain boxes on a shelf, you know there's stuff in them but not what. Semantic HTML elements are labeled boxes – "books," "tools," "games" – making it clear what's inside.

Semantic HTML improves accessibility, SEO, and general comprehension for both people and computers processing the text by utilizing tags like <header>, <footer>, <nav>, and <article>. Semantic HTML essentially promotes clarity and understanding by defining the function and connections between different parts of a webpage, which allows for more meaningful interactions with web material.

Q3.What is the role of the <head> section in an HTML document?

Answer -

The <head> section in an HTML document acts as the control center, providing essential information not directly displayed to users but crucial for the browser's understanding and presentation of the webpage. Think of it as the behind-the-scenes instructions for your web page.

Here's what the <head> section typically holds:

- **Document Title:** Defined by the <title> tag, this shows up in the browser tab and search engine results. It's essential for clarity and SEO (Search Engine Optimization).
- **Character Encoding:** Using the <meta> tag, you specify the character set used in your document, ensuring proper display of different languages and symbols.
- **Stylesheet Links:** The <link> tag allows you to connect external CSS files that define the page's styling, like fonts and colors.
- **JavaScript Inclusion:** Similar to stylesheets, you can embed or link to JavaScript files (.js) using the <script> tag. These scripts add interactivity to your webpage.
- **Other Metadata:** The <meta> tag offers a versatile way to include additional information for search engines, social media sharing, or browser behavior**. For instance, you can specify keywords relevant to your content or a description for social media previews.

Q4. What is the difference between the and <div> elements in HTML, and when should you use each?

Answer -

The main difference between and <div> elements in HTML boils down to their display and purpose:

Display:

<div>: Block-level element: It starts on a new line and stretches horizontally to fill the available space by default. Imagine it as a building block for your webpage content, creating distinct sections.

: Inline element: It flows with the surrounding text, only taking up the space required for its content. Think of it as inline highlighting within a sentence.

Purpose:

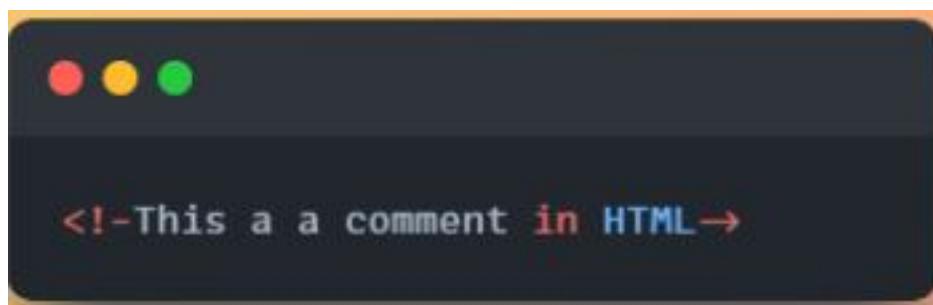
<div>: Used for major sections or content groupings, like headers, navigation, articles, or sidebars. It provides a clear structure for your webpage.

: Used for inline styling or grouping smaller elements within a line of text. Common uses include highlighting specific words, applying different fonts to a phrase, or wrapping inline elements like images or icons within text.

Q5. How comments are added in HTML?

Answer -

In HTML comments are created with the help of <!--...--> tag. Everything placed between them is a comment and it will not be displayed on the web.



INTERMEDIATE

Q6. What is the difference between HTML and HTML5?

Answer -

HTML	HTML5
HTML does not include semantic elements.	HTML5 includes semantic elements such as <header>, <footer>, <section>, and <article>.
<ul style="list-style-type: none"> • It didn't support audio and video without the use of Flash player support. • It does not allow JavaScript to run in the browser. • The doctype declaration is very long and complex. • It uses cookies to store temporary data. 	<ul style="list-style-type: none"> • It does not support audio and video with the help of <audio> and <video> elements. • It allows JavaScript to run in the background. This is possible due to JS Web worker API in HTML5. • Doctype is very simple in HTML5. <ul style="list-style-type: none"> • Uses SQL database and application cache to store data offline.

Q7.What are HTML Entities?

Answer -

HTML entities provide some methods that help us to display the reserved characters in the HTML or those characters that are not generally present on the keyboard such as '<' and (£, ¥, €, ©). HTML provides some Entity names and Entity numbers to use these symbols. Entity names are case-sensitive.

Syntax:

```
&entity_name; or &#entity_number;
```

Examples:

Notation	Symbols	Entity name	Entity Number
	non-breaking space	 	
<	less than	<	<

®	registered trademark	®	®
©	copyright	©	©
€	euro	€	€
¥	yen	¥	¥
£	pound	£	£
¢	cent	¢	¢

Q8.What is the purpose of the <meta> tag in HTML?

Answer -

In HTML, metadata about an HTML document is defined using the <meta> tag. In simple terms, metadata is information about the information on your webpage. Consider it as a label that indicates the contents of a file folder without requiring you to access it.

The <head> part of your HTML document contains <meta> tags, which give instructions to browsers and search engines without changing the look of the webpage itself. The following are some typical uses for <meta> tags:

Specifying character encoding: This tells the browser what character-set to use to interpret the text on your page.

Providing a page description: This is a brief summary of the content on your page, often used by search engines.

Defining keywords: While not as important as it once was, you can use <meta> tags to specify relevant keywords that describe the content of your page.

Setting the viewport: This controls how your webpage is displayed on different devices, ensuring it's mobile-friendly.

Author information: You can specify the author of the document.

Search engine instructions: You can use <meta> tags to instruct search engines on how to index your page (e.g., noindex to prevent indexing).

Q9.What is Doctype?

Answer -

An instruction that tells the web browser what version of HTML the document is written in is found at the very beginning of an HTML document and is called the DOCTYPE, or document type declaration.

The following are some important DOCTYPE points:

- Function: Its purpose is to guarantee that the webpage is rendered by the browser according to the correct standards for that specific version of HTML. This lessens the chance of inconsistent page displays across various browsers.
- History: DOCTYPE also referred to a Document Type Definition (DTD) in earlier versions of HTML (HTML 4.01 and below), which provided guidelines for the structure of the document. But HTML5 is not dependent on DTDs.
- As of Right Now: The DOCTYPE statement in HTML5 is less complicated and only indicates that the document is HTML:

<!DOCTYPE html>

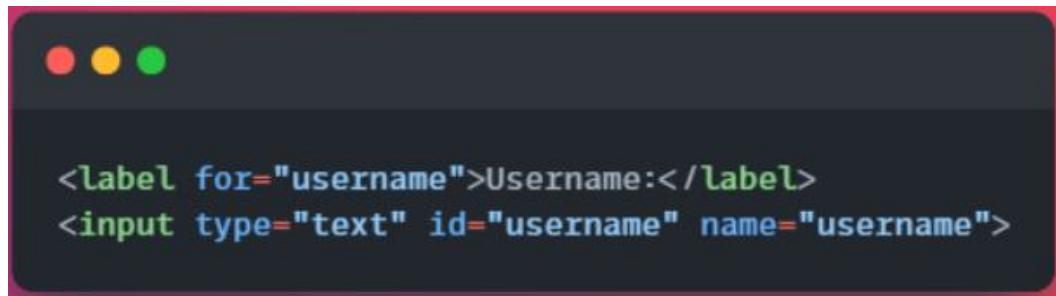
Importance: While not strictly mandatory in very basic cases, it's still considered good practice to include a DOCTYPE declaration in your HTML documents. It helps to ensure compatibility and avoids potential rendering issues.

Q10. Purpose of the "for" attribute with the label?

Answer -

The for attribute in HTML is used to link the label to a particular form control like a <textarea>, <select>, or <input>.

If a <label> element includes a "for" attribute, the form control it is meant to label's "id" attribute should match the value of that attribute. By clicking on the label, users can interact with the form control through this association, which can enhance accessibility and user experience.



```
<label for="username">Username:</label>
<input type="text" id="username" name="username">
```

Q11.What is the difference between block and inline elements?

Answer -

Inline Element	Block Element
Only the necessary amount of width is occupied by inline items.	No matter how big they are, block elements take up the entire width.
Inline elements don't start in a new line.	Block elements always start in a line.
Other inline elements can be positioned behind inline elements.	Block elements cannot have other components positioned behind them.
Elements that are inline lack upper and lower margins.	There are top and bottom margins for block elements.

Q12.Explain the differences between the HTML <button>, <input type="button">, and <input type="submit"> elements, and when to use each in form design.

Answer -

The HTML elements <button>, <input type="button">, and <input type="submit"> each have different behaviors and properties that serve comparable but different roles in form design.

1. <button>:

- A clickable button is created inside a form by using the <button> element. It may have other HTML elements, text, or images in it.
- <button> elements can include nested elements like , <i>, or and offer rich content, in contrast to <input> elements.
- CSS styling can be applied to the <button> element to change its look.
- It is frequently utilized when complex or personalized buttons—like those with stylized text or icons are needed.

2. <input type="button">:

- While it can make a clickable button inside a form, the element has fewer customization options than the <button> element. It is mostly used to generate simple buttons with text labels; it does not support nested elements.
- This kind of button is helpful for launching JavaScript functions or submitting simple forms.
- It is often used when the form's functionality can be satisfied with a simple, basic button design.

3. <input type="submit">:

- Since it similarly produces a clickable button inside a form, the element behaves differently in relation to form submission.
- A form with a button that is clicked sends the data to the server to be processed.
- Similar to this, it is usually used for simple form submissions and does not support nested components.
- When creating forms where submitting form data is the main activity, this is the best option.

ADVANCE

Q13.Describe the accessibility features and techniques in HTML for making web content more inclusive and usable by people with disabilities.

Answer -

To help to improve accessibility and guarantee that web content is inclusive and useable by individuals with impairments, HTML offers several features and strategies. The following are some essential HTML accessibility features and methods.

Semantic HTML Elements: Screen readers and other assistive devices are able to better understand the company and content of a webpage by using semantic HTML elements like <header>, <nav>, <main>, <article>, <section>, and <footer> .

Alternative Text for Images: By using the alt attribute in the element to provide descriptive alt attributes for images, screen readers may make sure that visually impaired people understand the purpose and substance of the images.

Form Accessibility: By labeling form elements with the element, you may make it easier for users to grasp the purpose of each form field by ensuring that screen readers associate form controls with their labels.

Accessibility and usability are enhanced when elements are explicitly associated with the form controls they correspond to by using the "for" property.

Accessible Links: Links that are accessible should have descriptive link language that makes it obvious where the link will lead. This will help people with disabilities understand the purpose of the link, especially if they are using a screen reader.

The usability and accessibility of links are enhanced when general wording like "click here" or "read more" is avoided.

Keyboard Accessibility: People who are unable to use a mouse can navigate and engage with the material using only their keyboard if all interactive components on the webpage, including buttons, links, and form controls, are keyboard accessible.

By giving interactive elements emphasis styles, keyboard users may quickly determine which element is in focus while navigating the webpage.

Headings and Semantic Structure: Content organized according to the recommended heading hierarchy (h1–h6) facilitates faster and more effective website navigation for people with disabilities, particularly when utilizing keyboard or screen reader navigation.

Making sure the header structure appropriately represents the hierarchy of material enhances usability and accessibility for all users.

Accessibility of Audio and Video: Ensuring users who are hard of hearing or deaf can access audio information is ensured by providing transcripts and captions for videos.

Including captions for audio files or descriptive text makes the content accessible to hard-of-hearing or deaf users.

Contrast in Color: Providing a sufficient contrast ratio between the background and text colors enhances the legibility of text for individuals with low vision or color vision impairments.

It is ensured that people with color vision deficits can still comprehend the text by avoiding the use of color alone to convey information.

Q14. What is the purpose of the HTML <iframe> element, and how can you use it to embed external content in a web page?

Answer –

The HTML <iframe> element serves as a versatile tool for integrating external content seamlessly into web pages, enhancing their functionality and interactivity. Let's delve into its purpose and usage with an illustrative example.

Consider that you are creating a trip blog and that you would like to provide dynamic maps of different locations. You choose to use iframes to incorporate Google Maps into your blog entries in order to accomplish this.



This example shows how to embed a Google Maps view of a location using the `<iframe>` element. The source URL of the embedded content, a Google Maps URL with embedded parameters is specified in the "src" attribute. The width and height properties also define the embedded map's dimensions. You may enable readers interactive maps to explore from within your webpage by putting this `<iframe>` code snippet in your travel-related blog entries. By providing context and visual assistance related to the places you're talking about, improves the user experience.

The `<iframe>` element is commonly used to achieve various goals, including:

- Embedding External Websites
- Displaying Maps
- Including Ads

Q15. When incorporating iframes in a web application, what potential security risks should be taken into consideration? Can you provide examples of how these risks can be exploited and suggest strategies or best practices to mitigate these security concerns?

Answer -

It's critical to be aware of any security threats that can jeopardize the integrity and safety of a web application when incorporating iframes. Among the main security issues with iframes are the following:

Clickjacking: It is the practice of deceiving visitors into clicking on a hidden iframe without their knowledge by superimposing an unseen iframe over a button or other legal website element. This may cause the user to do unexpected activities, such as revealing private information or carrying out illegal transactions.

Cross-Site Scripting (XSS): If an iframe permits the embedding of untrusted content without sufficient validation or sanitization, it may be used as a vector for XSS attacks. Malicious scripts can be injected by attackers into iframes, which are then run inside the hosting webpage's context, possibly compromising user data or session credentials.

Data leakage: Sensitive information from the parent webpage may unintentionally be exposed to the embedded content by iframes that load content from external sources. If the iframe is hosted on a separate website and has access to the properties or data of the parent window, this can happen.

When using iframes in a web application, take into consideration putting these security concerns under control by putting these best practices into practice:

Use Content Security Policy (CSP): To limit the sources from which iframes can load content, use CSP headers. This limits the execution of scripts and resources to trustworthy domains, hence preventing XSS attacks and clickjacking.

Put Frame-Busting Techniques into Practice: To stop your pages from loading inside iframes on malicious websites, use frame-busting tactics such as using JavaScript code or changing the X-Frame-Options header.

Verify and Clean the Input: To reduce the danger of XSS attacks, make sure that any dynamic or user-generated material loaded into iframes has been thoroughly inspected and sanitized. Make use of frameworks or libraries with XSS prevention techniques integrated in.

By implementing these security measures, web developers can mitigate the inherent risks associated with incorporating iframes into their web applications and ensure a more secure browsing experience for users.

16) What is the difference between block, inline, and inline-block elements in HTML/CSS?

Answer-

block: Takes up the full width available, and starts on a new line (e.g., `<div>`, `<p>`).

inline: Takes up only as much width as necessary, does not start on a new line (e.g., ``, `<a>`).

inline-block: Like inline, but respects width and height properties.