Topic: Interview Questions Set-1

1. **What is HTML? Differences between HTML & HTML5? Advantages of HTML5?**

   - **Answer:** HTML (HyperText Markup Language) is the standard markup language used to structure content on the web. HTML5, the latest version, adds semantic elements like <header>, <footer>, and <section>, improving readability and SEO. HTML5 also supports multimedia elements like <video> and <audio>, as well as APIs for offline storage, drawing (canvas), and geolocation. HTML5 reduces the need for third-party plugins, making web applications more efficient and accessible.

2. **What is the difference between HTML and XHTML?**

   - **Answer:** XHTML (eXtensible Hypertext Markup Language) is a stricter form of HTML that follows XML syntax rules. HTML is more forgiving with syntax errors, whereas XHTML requires all tags to be properly nested, closed, and written in lowercase. XHTML is case-sensitive and requires attributes to be quoted, making it more predictable and reliable for processing.

3. **What is the role of DOCTYPE in HTML?**

   - **Answer:** The <!DOCTYPE> declaration informs the browser about the version of HTML being used, allowing it to render the document consistently. In HTML5, <!DOCTYPE html> triggers standards mode, ensuring the page adheres to modern rendering standards instead of quirks mode (which accommodates older HTML practices).

4. **Difference between Head & Body in HTML? Where to place the JS link reference?**

   - **Answer:** The <head> tag contains metadata (like the page title, meta tags, and CSS links) which is not directly visible on the page, while the <body> contains the page's visible content. JavaScript links should be placed at the end of the <body> for faster page loading, as this prevents scripts from blocking the rendering of visible content.

5. **What is the Title Tag in HTML?**

   - **Answer:** The <title> tag sets the title of the webpage displayed in the browser tab and is also used by search engines for indexing. A descriptive title helps improve the page’s search engine ranking and provides context to users.

6. **What are different HTML heading tags?**
    - **Answer:** HTML provides six levels of headings, <h1> through <h6>, which represent headings in decreasing order of importance. <h1> is typically used for main headings, while <h6> is used for less significant headings. Proper use of headings improves accessibility and SEO by structuring content hierarchically.
7. **What are Meta Tags?**
    - **Answer:** Meta tags provide metadata about the HTML document, such as its description, author, keywords, and viewport settings. These tags help with SEO, defining how the page appears in search results, and control responsive behavior on different screen sizes.
8. **What are HTML Elements? What is the difference between Element & Tag?**
    - **Answer:** An HTML element includes the opening and closing tags along with any content within them. A tag refers to the markup symbols themselves, like <p> or </p>. For instance, <p>Hello World</p> is an HTML element with the tag <p> and content "Hello World."
9. **What are the roles and uses of the <div> element in HTML?**
    - **Answer:** The <div> element is a block-level container used for grouping content, which helps structure HTML documents for styling and layout purposes. It doesn’t add meaning to content, so it’s mainly used with CSS and JavaScript for layout control.
10. **What is the difference between <div> and <span> elements?**
    - **Answer:** <div> is a block-level element that starts on a new line and takes up the full width, whereas <span> is an inline element that only takes up as much space as necessary without creating a new line. Use <div> for larger sections of content and <span> for styling small parts of text within a line.
11. **What is the use of <label> Tag?**
    - **Answer:** The <label> tag is used in forms to provide labels for form elements. When associated with a form control, it enhances accessibility by making the label clickable, allowing users to activate the input by clicking on the label.

12. **What is the role of <a>, <br>, <hr>, <em>, <img>, <input>, & <button> elements?**

- **Answer:**
    - <a>: Creates hyperlinks to other web pages or resources.
    - <br>: Inserts a line break.
    - <hr>: Creates a horizontal line, typically used as a thematic break.
    - <em>: Emphasizes text, generally displayed in italics.
    - <img>: Embeds an image in the page.
    - <input>: Allows user input in forms (text, buttons, checkboxes, etc.).
    - <button>: Creates a clickable button, often used to submit forms.

13. **What is the role of <header>, <main>, <section>, <footer>, & <address> elements in HTML?**

- **Answer:** These semantic elements structure content more clearly:
    - <header>: Represents introductory content, like a logo or navigation links.
    - <main>: Encloses the main content unique to the page.
    - <section>: Defines standalone sections, such as a chapter or topic.
    - <footer>: Contains footer information, like copyright notices.
    - <address>: Provides contact information, often in the footer.

14. **What are Root, Parent, Child, & Nested elements?**

- **Answer:** Root elements are the top-level elements. Parent elements contain other elements within them (children), while child elements are enclosed within a parent. Nested elements are any elements inside other elements, creating a hierarchy.

15. **What are Empty Elements?**

- **Answer:** Empty elements do not have content or a closing tag, such as <br> or <img>. They are used where no wrapping content is needed, only the function or visual effect of the element.

16. **What are Semantic Elements in HTML?**

- **Answer:** Semantic elements like <article>, <section>, and <nav> provide meaning to the browser and developers, describing the content they enclose. They enhance accessibility and SEO by making the structure and purpose of content clearer.

17. **Can HTML tags be written in Uppercase?**

- **Answer:** HTML tags are case-insensitive, but lowercase is recommended for consistency and readability as per the HTML5 standard.

18. **What are the 3 differences between Block-Level & Inline Elements?**

- **Answer:**
    - Block-level elements (e.g., <div>, <p>) start on a new line and span the full width.
    - Inline elements (e.g., <span>, <a>) only occupy as much width as their content needs and remain within the line.
    - Block-level elements can contain both inline and other block elements, while inline elements generally cannot contain block-level elements.

19. **List all Block-Level & Inline Elements in HTML.**

- **Answer:**
    - Block-level: <div>, <p>, <ul>, <table>, <section>.
    - Inline: <span>, <a>, <img>, <strong>, <em>.

20. **What are HTML Attributes? What are the Types of HTML attributes?**

- **Answer:** HTML attributes provide additional information about elements, modifying their behavior or appearance. Types include:
    - **Global attributes:** id, class, style.
    - **Specific attributes:** e.g., href for <a>, src for <img>.

21. **What are the id, style, & class attributes of an element? When to use what?**

- **Answer:**
    - id: Uniquely identifies an element; use for specific elements needing unique styling or behavior.

    - class: Groups elements with similar styles; use for multiple elements needing the same CSS.
    - style: Applies inline CSS directly to an element; use sparingly for one-off styles.

22. **What will happen if two elements have the same ids?**
    - **Answer:** Having duplicate id values is invalid HTML, leading to unpredictable behavior in CSS and JavaScript targeting as each id is meant to be unique.
23. **What are Data Attributes in HTML?**
    - **Answer:** Data attributes, prefixed with data-, allow embedding custom data within elements for JavaScript access, often useful for dynamically generated content.
24. **What are the 5 Types of Links in HTML?**
    - **Answer:** Types include:
        - **Internal:** Link within the same page.
        - **External:** Links to other websites.
        - **Anchor:** Jumps to a specific section within a page.
        - **Email:** Opens the default mail client with a mailto: link.
        - **Phone:** Initiates a phone call using tel: on compatible devices.
25. **What is the difference between Absolute & Relative URLs?**
    - **Answer:**
        - Absolute URLs specify the full path (e.g., https://example.com/page), working independently of the current location.
        - Relative URLs specify paths relative to the current location (e.g., /page), simplifying internal site links.
26. **What is the purpose of the <nav> element in HTML?**
    - **Answer:** <nav> groups navigation links, allowing easy identification by assistive technologies and helping organize page structure semantically.
27. **How do you add an external stylesheet in your HTML?**

- **Answer:** Use the <link> tag inside <head>, setting the href to the CSS file's URL, providing consistent styling across multiple pages.

28. **How do you open a link in a new tab?**
    - **Answer:** Use target="_blank" in the <a> tag to open links in a new tab, often used for external links.
29. **How do you create an Email Link?**
    - **Answer:** Use mailto: in the href attribute of an <a> tag to open the default mail client with a pre-filled email address.
30. **What are the different Types of Lists in HTML?**
    - **Answer:** HTML supports ordered lists (<ol>), unordered lists (<ul>), and definition lists (<dl>), each structuring items differently.
31. **What is a Nested List in HTML?**
    - **Answer:** A nested list is a list within another list, which can be useful for hierarchical data, like categories and subcategories.
32. **What are <table>, <tr>, <th>, <td> elements?**
    - **Answer:**
        - <table>: Creates a table.
        - <tr>: Defines a row.
        - <th>: Creates a header cell.
        - <td>: Defines a data cell.
33. **What is the colspan attribute in HTML?**
    - **Answer:** colspan merges a cell across multiple columns, useful for spanning table headers or cells over multiple columns.
34. **What is the best way to add a border to a table, column, and cell?**
    - **Answer:** Use CSS properties like border, border-collapse, and border-spacing to customize table borders, achieving consistent styling.
35. **What is CSS? What are the 3 ways to Implement CSS in HTML?**
    - **Answer:** CSS (Cascading Style Sheets) styles HTML documents. Implementation methods:
        - **Inline CSS:** style attribute.

    - **Internal stylesheet:** <style> tag in <head>.
    - **External stylesheet:** <link> tag for linking an external CSS file.

36. **What is Inline Style in CSS? When to use it in real applications?**
    - **Answer:** Inline styles are written directly in an element's style attribute. They are generally used for quick fixes but avoided in large-scale applications due to maintainability issues.
37. **What is Internal Stylesheet in CSS? When to use it in real applications?**
    - **Answer:** Internal stylesheets are defined in <style> tags within <head>. They are best for single-page applications or testing specific page styles without affecting other pages.
38. **What is External Stylesheet in CSS? When to use it in real applications?**
    - **Answer:** External stylesheets link to CSS files, ideal for reusable styles across multiple pages, enabling better organization and reusability.
39. **What are CSS Selectors and what are their types?**
    - **Answer:** CSS selectors target HTML elements for styling. Types include:
        - **Element:** targets tags, e.g., p.
        - **Class:** targets classes, e.g., .class.
        - **ID:** targets unique IDs, e.g., #id.
        - **Attribute:** targets attributes, e.g., [type="text"].
        - **Pseudo-classes/elements:** target special states/elements, e.g., :hover, ::before.
40. **How do you Include CSS in a webpage or HTML?**
    - **Answer:** CSS can be added using inline styles, internal stylesheets, or external stylesheets linked with <link> in <head>.
41. **Explain Box Model in CSS.**
    - **Answer:** The box model describes element layout, comprising content, padding (space inside the element border), border, and margin (space outside the border), affecting spacing and positioning.
42. **Explain Padding, Margin, and Border.**

- **Answer:**
    - **Padding:** Space between the content and border.
    - **Border:** The line surrounding padding.
    - **Margin:** Space outside the border separating elements.

43. **What are the different data types available in JavaScript? Provide examples of each.**

- **Answer:** JavaScript data types include:
    - **String:** "hello"
    - **Number:** 5
    - **Boolean:** true
    - **Object:** {key: "value"}
    - **Array:** [1, 2, 3]
    - **Undefined:** variable without a value.

44. **Explain the difference between var, let, and const. When would you use each of these declarations?**

- **Answer:**
    - var: Function-scoped, can be redeclared; avoid in modern JavaScript.
    - let: Block-scoped, reassignable.
    - const: Block-scoped, non-reassignable; best for constants.

45. **What is hoisting in JavaScript? How does it affect variables declared with var, let, and const?**

- **Answer:** Hoisting moves variable/function declarations to the top of the scope. var is hoisted without initialization, let and const are hoisted but remain uninitialized, resulting in a Temporal Dead Zone if accessed before assignment.

46. **What is a callback function in JavaScript? Give an example of how and when to use a callback function.**

- **Answer:** A callback is a function passed as an argument to another function, executed after the parent function completes. Useful for handling asynchronous events, like data fetching or animations.

47. **What are arrow functions? How do they differ from traditional functions in JavaScript?**

- **Answer:** Arrow functions are concise expressions that do not bind this, making them useful for shorter functions and methods within classes.

48. **Explain the concept of 'callback hell' in JavaScript. Why does it occur, and what problems does it cause?**

- **Answer:** Callback hell occurs when many nested callbacks create complex, hard-to-read code. It complicates error handling and maintenance.

49. **What are some common solutions to avoid callback hell in JavaScript? Provide code examples for one solution.**

- **Answer:** Solutions include Promises, async/await, and modularizing functions. Promises allow chaining callbacks for better readability.

50. **Describe the scope of variables in JavaScript. How does the scope differ between variables declared with var, let, and const?**

- **Answer:**
    - var: Function-scoped.
    - let and const: Block-scoped, making them better for block-based scopes like loops and conditional statements.

51. **How does JavaScript handle asynchronous operations? Briefly explain the role of the event loop in this process.**

- **Answer:** JavaScript uses an event loop to handle async operations, checking the task queue for functions waiting to execute once the call stack is clear, enabling non-blocking code execution.

52. **What is the difference between synchronous and asynchronous code in JavaScript? Provide an example of each.**

- **Answer:**
    - Synchronous code executes sequentially, blocking until a task completes.
    - Asynchronous code allows multiple tasks to proceed without waiting, useful for time-consuming operations like API calls.