HTML CSS Interview Questions 40

01. What is the difference between and ?

Answer: is a block-level element that takes full width and starts on a new line, while is an inline element that stays within the line of text. Use to group or lay out sections, and for styling small chunks of content like a word or phrase inside a sentence.

02. What is the role of the alt attribute in an <img> tag?

Answer: The alt attribute provides alternative text when an image can't be displayed, and it's essential for screen readers. It improves accessibility for visually impaired users and boosts SEO by allowing crawlers to understand image context.

03. What is the difference between id and class attributes?

Answer: An id is unique and used once per page, while class can be reused across multiple elements. CSS uses #id and .class to select them, and id has higher specificity, making it ideal for targeting single elements precisely.

04. What is semantic HTML?

Answer: Semantic HTML uses meaningful tags like <header>, <main>, <article> to describe content structure. It improves accessibility and SEO by helping browsers and assistive technologies understand the purpose of each section.

05. What does the <!DOCTYPE html> declaration do?

Answer: It tells the browser to render the document using HTML5 standards instead of quirks mode. This ensures consistent behavior across modern browsers for layout and rendering.

06. What are meta tags used for?

Answer: meta tags provide metadata like character set, viewport settings, and SEO-related keywords. Though invisible to users, they are critical for browser rendering, mobile responsiveness, and search engine indexing.

07. What's the difference between <section>, <article>, and ?

Answer: <section> groups related content, <article> contains standalone content like a blog post, and is a non-semantic wrapper. Use semantic tags for meaningful structure, which improves accessibility and SEO.

08. What is the href attribute used for?

Answer: The href attribute defines the URL a link (<a>) or stylesheet (<link>) points to. It tells the browser where to navigate or what resource to fetch when the element is triggered.

09. What does target="_blank" do?

Answer: It opens the linked document in a new browser tab or window. This is especially useful when linking to external sites so users don't leave your site.

10. What's the difference between <input type="text"> and <textarea>?

Answer: <input type="text"> is designed for single-line text, like names or emails, while <textarea> handles multi-line input like messages or comments. Textareas support scrolling and line breaks; input fields don't.

11. What is the action attribute in a form?

Answer: The action attribute defines the URL where the form's data will be sent after submission. It's typically used with the method attribute (GET or POST) to perform server-side operations.

12. What is the purpose of the <label> tag?

Answer: <label> provides a user-friendly name for form inputs and improves accessibility. When linked via for and id, clicking the label focuses the corresponding input field.

13. What is the difference between required, disabled, and readonly?

Answer: required ensures a field must be filled before submission, disabled makes the input unusable and skips it during submission, and readonly allows viewing but not editing while still submitting the value. Each modifies user interaction and submission behavior.

14. What does rel="noopener noreferrer" do?

Answer: It improves security when using target="_blank" by preventing the new page from accessing the original page via window.opener. This protects against reverse tab-nabbing attacks.

15. Can a form be submitted without a submit button?

Answer: Yes, forms can be submitted using JavaScript (form.submit()) or by pressing the Enter key in an input field. This enables more dynamic or custom submission behavior without a visible button.

16. What is the <noscript> tag?

Answer: The <noscript> tag displays fallback content if JavaScript is disabled in the browser. It ensures users without JS support still receive essential information or guidance.

17. What is autocomplete in input fields?

Answer: autocomplete helps users quickly fill in inputs by suggesting previously entered values stored in the browser. This enhances user experience and speeds up form completion.

18. What's the difference between placing <script> in <head> vs. before </body>?

Answer: Placing <script> at the end of <body> allows HTML to load first, improving page performance. Scripts in <head> block rendering unless marked with async or defer.

19. What does the hidden attribute do?

Answer: It hides the element from view but keeps it in the DOM, acting like display: none;. It's useful for toggling content visibility via JavaScript.

20. What are data attributes?

Answer: Data attributes (data-*) allow you to store extra info on HTML elements without affecting layout or semantics. They're useful for custom behavior in JavaScript and accessed using element.dataset.

21. What is the difference between `relative`, `absolute`, `fixed`, and `sticky` positioning in CSS?

Answer: `relative` moves element from its normal position. `absolute` positions relative to nearest positioned ancestor. `fixed` stays relative to viewport. `sticky` toggles between relative and fixed based on scroll position.

22. What is specificity in CSS and how is it calculated?

Answer: Specificity determines which rule is applied when multiple rules match. It's calculated using a point system: inline styles > IDs > classes > elements. More specific rules override less specific ones.

23. What's the difference between `em`, `rem`, `%`, and `px` units?

Answer: `px` is absolute. `em` is relative to the element's font-size. `rem` is relative to root (`html`) font-size. `%` is relative to parent element's value.

24. What is the box model in CSS?

Answer: The box model consists of `content`, `padding`, `border`, and `margin`. Understanding how each layer adds to total size helps in layout calculations.

25. What is the difference between `visibility: hidden` and `display: none`?
Answer: `visibility: hidden` hides the element but keeps its space. `display: none` removes it from the layout entirely.

26. How does `z-index` work and when does it fail?
Answer: `z-index` controls stacking order of elements. It only works on positioned elements (relative, absolute, fixed). If parent doesn't have positioning, it may behave unexpectedly.

27. What is the difference between `min-width`, `max-width`, and `width`?
Answer: `width` sets fixed size. `min-width` ensures the element doesn't shrink below a value. `max-width` ensures it doesn't grow beyond a value. Useful in responsive design.

28. Explain the concept of Flexbox in CSS.
Answer: Flexbox is a layout model that arranges items in rows or columns. It offers alignment, spacing, and distribution capabilities without using float or positioning.

29. Difference between `auto`, `inherit`, `initial`, and `unset` values?
Answer: `auto` uses browser default. `inherit` takes value from parent. `initial` resets to default CSS spec. `unset` behaves like `inherit` for inherited properties and `initial` for others.

30. How do media queries work in CSS?
Answer: Media queries apply styles based on device characteristics like width, height, or orientation. They enable responsive design by targeting different screen sizes.

31. What is the difference between `inline`, `block`, and `inline-block` elements?
Answer: `inline` flows with text but doesn't accept height/width. `block` breaks line and accepts box model. `inline-block` acts like inline but respects height/width.

32. What are pseudo-classes and pseudo-elements in CSS?
Answer: Pseudo-classes (e.g., `:hover`, `:nth-child()`) style elements based on state or position. Pseudo-elements (e.g., `::before`, `::after`) create virtual elements in the DOM for styling.

33. How does the `calc()` function work in CSS?

Answer: `calc()` allows dynamic calculations in CSS (e.g., `width: calc(100% - 60px)`). Useful for flexible layouts that require arithmetic.

34. What are CSS variables and why use them?
Answer: CSS variables (`--primary-color`) allow reuse of values across stylesheets. They improve maintainability and support dynamic theming with JavaScript.

35. What's the difference between `transition`, `transform`, and `animation` in CSS?
Answer: `transition` animates changes on hover/focus. `transform` changes appearance (rotate, scale). `animation` provides multi-step keyframe sequences.

36. What is the stacking context and how is it created?
Answer: A stacking context is a 3D space for elements' rendering order. It's created by elements with `position` and `z-index`, or CSS properties like `opacity < 1` or `transform`.

37. What's the difference between `overflow: auto`, `scroll`, `hidden`, and `visible`?
Answer: `auto` shows scrollbars only when needed. `scroll` always shows them. `hidden` clips content. `visible` shows overflow without scrollbars.

38. What is a pseudo-element and how is it used?
Answer: Pseudo-elements like `::before` and `::after` insert content before/after an element. They're useful for styling without adding extra DOM nodes.

39. How to center a div both horizontally and vertically?
Answer: Use Flexbox: `display: flex; justify-content: center; align-items: center;` on the parent. Alternatively, use absolute positioning with `transform: translate(-50%, -50%)`.

40. How does the `object-fit` property work?
Answer: `object-fit` defines how media like images/videos should resize within a container. Values like `cover`, `contain`, and `fill` control the fit behavior.