Top 20 CSS Interview Questions & Answers

01. 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.

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

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

03. Whats 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.

04. 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.

05. 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.

06. 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.

07. What is the difference between `min-width`, `max-width`, and `width`?

Answer: `width` sets fixed size. `min-width` ensures the element doesnt shrink below a value. `max-width` ensures it doesnt grow beyond a value. Useful in responsive design.

Top 20 CSS Interview Questions & Answers

08. 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.

09. 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.

10. 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.

11. What is the difference between `inline`, `block`, and `inline-block` elements?

Answer: `inline` flows with text but doesnt accept height/width. `block` breaks line and accepts box model. `inline-block` acts like inline but respects height/width.

12. 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.

13. 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.

14. 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.

15. Whats 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.

16. What is the stacking context and how is it created?

Answer: A stacking context is a 3D space for elements' rendering order. Its created by elements with `position` and `z-index`, or CSS properties like `opacity < 1` or `transform`.

17. Whats 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.

18. 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.

19. 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%)`.

20. 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.