

■ Center Div Flexbox Q&A;

■ HTML Structure & Semantics

Q: Why start with `<!DOCTYPE html>`?

A: It ensures the page is rendered in standards-compliant mode (HTML5), avoiding quirks.

Q: What does `lang="en"` do?

A: Declares document language, aiding screen readers and SEO.

Q: Roles of `<head>` & `<body>`?

A: `<head>` holds metadata/scripts; `<body>` contains visible content.

Q: Why include `<meta charset="UTF-8">`?

A: Sets UTF-8 encoding to support all characters.

Q: Purpose of viewport meta tag?

A: Controls layout scaling on mobile for responsiveness.

■ CSS Fundamentals

Q: Selector specificity order?

A: Inline > IDs > classes/attributes > elements; combining increases specificity.

Q: Difference between `px`, `%`, `vh`, `dvh`?

A: `px` absolute; `%` relative to parent; `vh` viewport height; `dvh` actual visible viewport height.

Q: How `%` sizing works inside nested flex?

A: Percentages refer to the flex container's content box dimensions.

Q: What is the CSS box model?

A: Content + padding + border + margin; use `box-sizing: border-box;` to include padding/border in size.

■ Layout with Flexbox

Q: What does `display: flex;` accomplish?

A: Turns a container into flex layout, enabling axis-based alignment.

Q: `align-items` vs `justify-content`?

A: `align-items` aligns on cross-axis; `justify-content` on main-axis.

Q: Default flex properties?

A: `flex-grow:0`, `flex-shrink:1`, `flex-basis:auto`; override to control sizing.

Q: Why use nested flex containers?

A: To independently align child elements at different levels.

Q: Old centering tricks vs Flexbox?

A: Older hacks include `absolute+transform` or `table-cell`; Flexbox is cleaner and more robust.

■ Viewport Units & Responsiveness

Q: Why use `100dvh` over `100vh`?

A: `dvh` reflects actual visible viewport, avoiding mobile address-bar jumps.

Q: How does layout adapt on rotation?

A: Units recalculate on orientation change, preserving proportions.

Q: Fallback for `dvh`?

A: Use `100vh` or update a CSS variable via JS on `resize` event.

■ Best Practices & Optimization

Q: How to DRY repeated flex styles?

A: Create a utility class: `.center-flex { display:flex; align-items:center; justify-content:center; }`.

Q: Naming conventions (BEM)?

A: Example: `.parent__outer-box`, `.parent__inner-box--third` for clarity.

Q: Performance considerations?

A: Deep nesting increases layout cost; keep structure as flat as possible.

Q: Preprocessors & utilities?

A: Use Sass mixins or Tailwind CSS for maintainability.

■ Comparisons & Additional Angles

Q: Flexbox vs Absolute centering?

A: Flex is dynamic/responsive; absolute is precise but less flexible.

Q: Flex vs CSS Grid?

A: Grid handles two-dimensional layouts; Flex excels on one axis.

Q: Accessibility with flex layouts?

A: Maintain logical DOM order and use ARIA roles if needed.

■ Cross-Browser & Future-Proofing

Q: Which browsers support `dvh`?

A: Modern browsers; check caniuse.com and provide fallbacks.

Q: Progressive enhancement strategy?

A: Let new features apply and gracefully degrade on older browsers.