

Interview Questions on HTML, CSS, Bootstrap, and JavaScript

HTML –

Q: What is the difference between `<div>` and `<section>` tags?

A: `<div>` is a generic container with no semantic meaning, used for grouping and styling. `<section>` is a semantic tag that defines sections of content with specific meaning, improving SEO and accessibility.

Q: What are data-* attributes and how are they used?

A: `data-*` attributes store custom data private to the page or application. Example: `<button data-user-id='123'></button>` accessed in JS as `element.dataset.userId`.

Q: Can a page have multiple `<header>` or `<footer>` tags?

A: Yes. You can have multiple `<header>` or `<footer>` tags inside different `<section>` or `<article>` elements.

Q: What is the difference between inline, block, and inline-block elements?

A: `block` takes full width and starts on a new line. `inline` only takes the necessary width. `inline-block` behaves like inline but allows width and height.

Q: What is the use of meta tags in HTML?

A: Meta tags provide metadata like character encoding, viewport, and SEO. Example: `<meta name='viewport' content='width=device-width, initial-scale=1.0'>`.

Q: What are semantic elements in HTML?

A: Semantic elements clearly describe their meaning. Examples include `<article>`, `<section>`, `<nav>`, and `<aside>`.

Q: What is the difference between HTML4 and HTML5?

A: HTML5 introduced semantic tags, multimedia support, APIs like local storage, and removed deprecated tags like `<font>`.

CSS –

Q: What is the difference between em, rem, %, and px in CSS?

A: `px` is absolute. `em` is relative to parent font size. `rem` is relative to root font size. `%` is relative to parent element size.

Q: What are pseudo-classes in CSS? Give examples.

A: Pseudo-classes define a special state. Examples: `:hover`, `:focus`, `:nth-child(n)`.

Q: Explain specificity in CSS.

A: Specificity is a hierarchy that determines which rule applies. Inline > ID > Class > Tag. Calculated as (1000,100,10,1).

Q: What is the box model in CSS?

A: The box model includes: Content → Padding → Border → Margin.

Q: What is the difference between relative, absolute, and fixed positioning?

A: `relative` moves from its normal position. `absolute` is relative to the nearest positioned ancestor. `fixed` is relative to the viewport.

Q: What is z-index and how does it work?

A: `z-index` sets the stack order. Higher values appear in front.

Q: How to center an element horizontally and vertically using Flexbox?

A: Use `display: flex; justify-content: center; align-items: center; height: 100vh;` on the container.

Bootstrap –

Q: What are Bootstrap's breakpoints and how do they work?

A: Breakpoints define screen size thresholds (`sm`, `md`, `lg`, etc.). Used in responsive classes like `col-md-6`.

Q: Difference between `.d-none` and `.invisible` in Bootstrap?

A: `.d-none` hides the element and removes it from layout. `.invisible` keeps space but hides the element.

Q: How does Bootstrap grid system work?

A: Uses a 12-column system. Classes like `col-md-6` divide space. Nest inside `.row` and `.container`.

Q: What is the difference between `.container` and `.container-fluid`?

A: `.container` has fixed width. `.container-fluid` spans 100% width.

Q: What are utility classes in Bootstrap?

A: Utility classes handle spacing, text alignment, display, etc., like `m-3`, `text-center`, `d-flex`.

Q: Explain the use of responsive display utilities in Bootstrap.

A: Classes like ``d-none d-md-block`` show/hide elements at different breakpoints.

Q: How does the Collapse component work in Bootstrap?

A: Used to toggle visibility. Uses ``data-bs-toggle='collapse'`` and class ``collapse``.

JavaScript –

Q: What is the difference between var, let, and const?

A: ``var`` is function-scoped and hoisted. ``let`` and ``const`` are block-scoped. ``const`` cannot be reassigned.

Q: What is event delegation in JavaScript?

A: Uses a parent to handle events of child elements via bubbling. Useful for dynamic elements.

Q: Explain the concept of closures in JavaScript.

A: A closure is a function that remembers the variables in its lexical scope even when called outside.

Q: What are template literals?

A: Introduced in ES6 using backticks. Allows string interpolation and multi-line strings.

Q: What are arrow functions and how are they different from regular functions?

A: Arrow functions have shorter syntax and do not bind ``this``.

Q: What is the difference between synchronous and asynchronous code?

A: Synchronous code runs line-by-line. Asynchronous code (like ``setTimeout``, ``fetch``) does not block execution.

Q: What is a Promise in JavaScript?

A: A Promise represents an async operation that can succeed or fail. Uses ``then()`` and ``catch()``.

Q: Explain the difference between == and ===.

A: ``==`` checks value with type coercion. ``===`` checks value and type.

Q: What is hoisting in JavaScript?

A: Variable and function declarations are moved to the top of their scope. Only declarations are hoisted, not initializations.

Q: Explain how this keyword works in JavaScript.

A: `this` refers to the current context. In arrow functions, `this` is inherited from the parent scope.

Q: What are higher-order functions in JavaScript?

A: Functions that take another function as an argument or return a function. Examples: `map`, `filter`, `reduce`.