Top Frontend Interview Questions & Answers

HTML:

What is the purpose of the doctype declaration?

The <!DOCTYPE> declaration defines the document type and version of HTML. It helps the browser render the page correctly.

Explain the difference between and .

 is a block-level element, while is an inline element used for smaller pieces of content.

CSS:

What is the box model in CSS?

The box model in CSS includes content, padding, border, and margin.

What are pseudo elements and classes?

Pseudo-elements style specific parts of elements, while pseudo-classes style elements based on their state.

Difference between 'visibility: hidden' and 'display: none'.

'visibility: hidden' hides the element but maintains space, while 'display: none' removes it from the layout.

Explain z-index.

z-index controls the vertical stacking order of elements in CSS.

JavaScript Basics:

Explain the concept of hoisting in JavaScript.

Hoisting moves variable and function declarations to the top of their scope before code execution.

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

let and const are block-scoped, while var is function-scoped.

Difference between == and ===?

== compares values with type conversion, while === compares both value and type.

Explain Event loop.

The event loop allows JavaScript to handle asynchronous operations like callbacks and promises.

JavaScript Functions:

What is a closure in JavaScript?

A closure is a function that has access to its outer scope, even after the outer function has returned.

Explain the differences between arrow functions and regular functions.

Arrow functions have lexical scoping for this and cannot be used as constructors.

Explain currying.

Currying transforms a function into a series of functions that each take one argument.

Explain Promises.

Promises represent asynchronous operations, allowing chaining with .then() and .catch().

Responsive Design:

What is the importance of media queries in responsive design?

Media queries allow you to apply styles based on screen size, improving responsiveness.

Describe the difference between em and rem units in CSS.

em is relative to the parent element?s font size, while rem is relative to the root element?s font size.

CSS Flexbox:

What is the flexbox model, and how does it work?

Flexbox is a layout system that organizes items along one axis, allowing flexible item sizing.

Explain the purpose of justify-content and align-items in flexbox.

justify-content aligns items along the main axis, while align-items aligns them along the cross axis.

CSS Grid:

How does CSS Grid differ from Flexbox?

CSS Grid is two-dimensional, handling both rows and columns, while Flexbox is one-dimensional.

Explain the use of the grid-template-columns property.

grid-template-columns defines the number and width of columns in a grid.

React Basics:

What is JSX in React?

JSX is a syntax extension that allows writing HTML-like structures in JavaScript.

Explain the purpose of state in React components.

State holds dynamic data that changes over time and re-renders the component when updated.

How to pass data from Parent to Child component and vice-versa?

Parent to child via props, child to parent via callback functions.

Explain the virtual Dom concept.

The virtual DOM is a lightweight copy of the actual DOM that allows efficient updates.

React Components:

Differentiate between functional and class components in React.

Functional components are stateless, while class components can hold state and have lifecycle

methods.

Describe the lifecycle methods in a React class component.

Common methods include componentDidMount, componentDidUpdate, and componentWillUnmount.

How can we achieve lifecycle methods in functional component?

Use the useEffect hook in functional components.

Difference between controlled and uncontrolled components.

Controlled components have their input values controlled by React state, while uncontrolled components rely on DOM.

What are pure components?

Pure components only re-render when their props or state change, improving performance.

React Hooks:

Explain the use of the useState hook in React.

useState allows functional components to manage state and trigger re-renders.

What is the useEffect hook, and why is it used?

useEffect handles side effects like data fetching in functional components.

What is the difference between useCallback and useMemo hook?

useCallback memoizes functions, while useMemo memoizes values or computations.

Explain useContext hook.

useContext allows components to access context values without prop drilling.

Redux:

What problem does Redux solve in a React application?

Redux manages state across a large application, allowing state to be accessed globally.

Explain the roles of actions, reducers, and the store in Redux.

Actions describe changes, reducers update the state, and the store holds the application?s state.

Explain useReducer.

useReducer is a React hook that manages state using a reducer function, similar to Redux?s reducers.