Frontend Developer Interview Questions and Answers

Q: What are the falsy values in JavaScript?

A: - false

- 0

- "" (empty string)

- null

- undefined

- NaN

Q: What are the differences between const, let, and var?

A: - const: Block-scoped, cannot be reassigned, not hoisted.

- let: Block-scoped, can be reassigned, not hoisted.

- var: Function-scoped, can be reassigned, hoisted.

Q: What is scoping in JavaScript?

A: - Function Scope: Variables accessible within a function.

- Block Scope: Variables accessible within a block ({}).

- Lexical Scope: Variables accessible in their defining scope and nested scopes.

Q: How does == vs === differ?

A: - ==: Checks value equality with type coercion.

- ===: Checks strict equality without type coercion.

Q: What is the difference between undefined and null?

A: - undefined: Declared but not assigned a value.

- null: Assigned value representing no value.

Q: What are some JavaScript data types?

A: - Number

- String

- Boolean

- Object

- Undefined

- Null

- Symbol (ES6)

- BigInt (ES2020)

Q: What do the spread and rest operators do?

A: - Spread (...): Expands an iterable into elements.

- Rest (...): Collects multiple elements into an array.

Q: What are some ES6 features used in JavaScript?

A: - Arrow functions

- Classes

- Template literals

- Destructuring

- Default parameters

- Rest and spread operators

- Promises

- Modules

- let and const

Q: Which array methods are ES6 features?

A: - find()

- findIndex()

- includes()

- some()

- every()

- fill()

- from()

- of()

Q: What is and why might you destructure an object or array?

A: - Destructuring: Unpacks values/properties into variables.

- Why: Simplifies code and improves readability.

Q: What is Git? Why do you use it? What are some common commands you use?

A: - Git: Version control system.

- Why: Manage code versions, collaborate, and maintain history.

- Common Commands: git init, git clone, git add, git commit, git push, git pull, git branch, git merge

Q: What is Agile? What is Scrum?

A: - Agile: Iterative software development methodology.

- Scrum: Agile framework focusing on iterative progress through sprints.

Q: What are some CSS selectors and how do they differ?

A: - Element Selector: Selects by tag name.

- Class Selector: Selects by class.

- ID Selector: Selects by ID.

- Attribute Selector: Selects by attribute.

- Pseudo-class Selector: Selects by state or position.

Q: What is specificity in CSS?

A: - Specificity: Determines which CSS rule is applied when multiple rules match the same element.

Q: What is responsive design and how might you implement it?

A: - Responsive Design: Ensures web pages look good on all devices.

- Implementation: Media queries, flexible grids, responsive images.

Q: What are some issues with numbers in JavaScript you may have to protect against?

A: - Floating-point precision issues.

- Handling large/small numbers.

- Dealing with NaN.

Q: What are Promises and why might you use them?

A: - Promises: Represent completion/failure of an async operation.

- Why: Cleaner async handling, avoids callback hell.

Q: What is pass by reference and how does it differ from pass by value?

A: - Pass by Value: Copies actual value; changes don't affect the original.

- Pass by Reference: Copies reference; changes affect the original.

Q: How do you stay up to date?

A: - Follow blogs/websites.

- Join online communities.

- Attend conferences/meetups.

- Take courses/tutorials.

Q: Why should we hire you?

A: - Experienced with modern frontend technologies.

- Proven track record in delivering high-quality interfaces.

- Strong problem-solving and collaboration skills.

Q: Why do you wanna work for us?

A: - Your innovative projects and culture align with my goals.

- Opportunity to work with a talented team and contribute meaningfully.

Q: What is your biggest professional achievement?

A: - Leading a project to redesign a major client's website, increasing user engagement by 25% and conversion rates by 15%.

Q: What is Virtual DOM in React? What is the difference between Virtual DOM and 'real' DOM?

A: - Virtual DOM: Lightweight in-memory representation of the real DOM.

- Difference: Virtual DOM updates are faster and update only differences to the real DOM.

Q: What is JSX?

A: - JSX: Syntax extension for JavaScript that resembles XML/HTML, used to describe UI in React.

Q: What is the difference between an element and a component?

A: - Element: A plain object describing UI.

- Component: A function or class returning elements for the UI.

Q: Can you write React without JSX?

A: - Yes, by using React.createElement() to create elements instead of JSX.

Q: How do you pass a value from parent to child? What about from child to parent?

A: - Parent to Child: Pass props.

- Child to Parent: Pass a callback function as a prop.

Q: What is prop-drilling?

A: - Prop-drilling: Passing data through multiple component layers via props.

Q: Can you modify props?

A: - No, props are read-only.

Q: What is the difference between props and state?

A: - Props: Data passed from parent.

- State: Internal data managed within a component.

Q: What is the component lifecycle?

A: - Sequence of methods invoked during a component's mounting, updating, and unmounting phases.

Q: How do you update the lifecycle in function components?

A: - Use React hooks like useEffect, useState, etc.

Q: What parameters does useEffect take in?

A: - A function to execute.

- Optional dependency array.

Q: When does the useEffect function run?

A: - After initial render and whenever dependencies change.

Q: What is the useEffect function's return value?

A: - An optional cleanup function for unmounting or re-running the effect.

Q: What is the difference between refs and state changes?

A: - Refs: Store mutable values without triggering re-renders.

- State: Triggers re-renders when updated.

Q: When is the best time to use refs?

A: - When interacting with the DOM or storing values that don't require re-renders.

Q: What is a proper way to update a ref in a function component?

A: - Use useRef and modify the current property directly.

Q: What is the difference between the Context API and prop drilling?

A: - Context API: Passes data through the tree without manual props.

- Prop Drilling: Manually passing props down multiple layers.

Q: When shouldn't you use the Context API?

A: - For infrequently changing data or data needed only by a few components.

Q: What is a Fragment?

A: - A wrapper to group multiple elements without adding extra DOM nodes.

Q: When should you use a Class-based component vs a function-based component?

A: - Class-based: For lifecycle methods pre-hooks.

- Function-based: Preferred for hooks and cleaner code.

Q: What is a higher-order component?

A: - A function taking a component and returning a new component with additional behavior.

Q: What is a portal?

A: - A way to render children into a DOM node outside the parent component's DOM hierarchy.

Q: What are uncontrolled components? What are controlled components?

A: - Uncontrolled: Inputs managed by the DOM, accessed via refs.

- Controlled: Inputs managed by React state, updated via state changes.