

Frontend Interview Preparation

1. Basic Web Concepts
 - a. Page rendering cycle
 - b. http/https/https2
 - c. CORS
 - d. Local storage/Session storage
 - e. Web Vitals
 - f. Cookie
 - g. JWT
 - h. XHR
 - i. Micro Frontend
 - j. REST/GraphQL/Socket connection
 - k. Browser Concepts
 - l. Debugging Application
 - m. Chrome Dev Tool Features
2. HTML
 - a. Semantic HTML5 Elements
 - b. HTML Forms and Validation
 - c. Accessibility (A11y)
 - d. Responsive Design with HTML and CSS
 - e. HTML APIs and Integrations
 - f. HTML Templates and Shadow DOM
 - g. Microdata and Schema.org
 - h. Performance Optimization
 - i. Forms and Accessibility
 - j. Web Security Fundamentals
3. CSS
 - a. Class and other selectors
 - b. Pseudo Classes
 - c. Box Model
 - d. Pseudo Elements
 - e. CSS type - flex, grid, normal
 - f. How to centre
 - g. pseudo classes and elements
 - h. All element states - active, hover
 - i. Media queries
 - j. Pre-processors - SCSS or LESS
 - k. mixins
 - l. CSS constants
 - m. BEM
 - n. Import
4. Javascript
 - a. Basic
 - i. Data types
 - ii. Functions
 - iii. Scope in JavaScript
 - iv. Closure

- v. Event loop
- vi. Prototype and prototype chain
- vii. Class and inheritance
- viii. DOM
- ix. bind/call/apply
- x. Promise
- xi. WebAPI
- xii. Task queue
- xiii. Call stack
- xiv. Async/await
- xv. Generators
- xvi. Typescript

b. Advance

- i. OOPs concept
- ii. Design Patterns
 - 1. Singleton
 - 2. Provider
 - 3. Prototype
 - 4. Observer
 - 5. Module
 - 6. HOC
- iii. Understanding V8 in depth
 - 1. JIT
 - 2. Interpreter
 - 3. Execution
 - 4. Compiler
- iv. Currying

5. React

a. eBook (100 interview ques) -

<https://drive.google.com/file/d/1QDqDFytObucUS5z7mAv20qMSSjByUuzw/view?usp=sharing>

b. Basic

- i. Introduction JSX
- ii. React Component
- iii. Component State and Props
- iv. Adding Style (CSS)
- v. Functional and Class components
- vi. React Lifecycle Methods
- vii. Virtual DOM
- viii. React Hooks
- ix. Custom Hooks
- x. Context API
- xi. Synthetic Events
- xii. Routing
- xiii. Data Flow (Redux/Flux)
- xiv. Server-Side Rendering
- xv. Unit Testing
- xvi. Jest & React Testing library
- xvii. Mocking Data

xviii. Understanding Webpack (Bundler)

xix. Babel, env, prettier, linter

6. Redux

- a. What are the core principles of Redux?
- b. How does Redux differ from the Context API in React?
- c. Can you explain the role of middleware in Redux and name some commonly used middleware?
- d. What is the purpose of combineReducers in Redux?
- e. Describe the steps of a typical Redux data flow.
- f. How do you handle asynchronous actions in Redux?
- g. What is the significance of immutability in Redux, and how do you ensure state immutability?
- h. What are selectors in Redux, and why are they useful?
- i. Can you provide an example of how to structure a large-scale Redux application?
- j. How do you debug a Redux application?