Javascript Interview Topics

- **Core Javascript**
  1. Variables and Scoping
     a. var, let, const
     b. Hoisting
     c. Data Types and Structures
     d. Temporal Dead Zone
  2. Primitive vs. reference types
     a. Arrays and objects
     b. Type coercion and type checking
     c. Functions
  3. Function declarations vs. expressions
     a. Arrow functions
     b. Closures
     c. IIFE (Immediately Invoked Function Expressions)
     d. Higher-order functions
     e. Asynchronous JavaScript
  4. Callbacks
     a. Promises
     b. async/await
     c. Event loop and microtasks/macrotasks
     d. Object-Oriented Programming (OOP)
  5. Prototypal inheritance
     a. ES6 classes
     b. this keyword, call, apply, bind
     c. Functional Programming
  6. Pure functions
     a. Immutability
     b. Function composition
     c. Array methods (map, filter, reduce)

- **Advance Javascript**
  7. Event Handling
     a. Event listeners
     b. Event propagation (bubbling and capturing)
     c. Event delegation
  8. Error Handling
     a. try, catch, finally
     b. Custom errors
  9. JavaScript Engine and Runtime
     a. V8 Engine
     b. Just-In-Time (JIT) compilation

       c. Memory management
10. Modules
       a. ES6 modules (import/export)
       b. CommonJS
11. Advanced Data Structures
       a. Sets and Maps
       b. WeakSet and WeakMap
12. Design Patterns
       a. Singleton
       b. Factory
       c. Module
       d. Observer
13. Concurrency and Parallelism
       a. Web Workers
       b. Service Workers

- **Frontend-Specific Topics**
  14. DOM Manipulation
         a. Selecting elements
         b. Modifying elements
         c. Event handling
  15. Web APIs
         a. try, catch, finally
         b. Custom errors
  16. JavaScript Engine and Runtime
         a. Fetch API
         b. WebSockets
         c. LocalStorage, SessionStorage, and IndexedDB
         d. Navigator API
  17. Frameworks/Libraries
         a. React (hooks, lifecycle methods, state management)
         b. Angular (directives, services, dependency injection)
         c. Vue.js (reactivity, components, Vuex)
  18. Performance Optimization
         a. Debouncing and throttling
         b. Lazy loading
         c. Code splitting

- **Testing**
  19. Unit Testing
         a. Jest
  20. Integration Testing
         a. Cypress
         b. Selenium

- **Miscellaneous**
  21. Version Control
      a. Git (branching, merging, rebasing)
  22. Build Tools
      a. Webpack
      b. Babel
      c. ESLint
  23. Package Management
      a. npm
      b. Yarn

- **Best Practices**
  24. Code Quality
      a. Clean code principles
      b. Code reviews
      c. Refactoring
  25. Security
      a. XSS, CSRF
      b. CORS
      c. Secure coding practices
  26. Accessibility
      a. ARIA roles
      b. Semantic HTML