

JavaScript Complete Roadmap (Basic to Advanced)

1. JavaScript Foundations (Basics)

- Basics & Syntax: var, let, const, hoisting, data types, type coercion, operators
- Control Structures: if-else, switch, loops (for, while, do...while, for...in, for...of)
- Functions: declarations, expressions, arrow functions, default/rest/spread parameters
- DOM Basics: selectors, events, bubbling, delegation
- ES6+ Basics: template literals, destructuring, modules

2. Intermediate JavaScript

- Objects: creation, methods (keys, values, entries), prototypes, 'this'
- Arrays: mutating/non-mutating methods, iterators, shallow vs deep copy
- Functions Deep Dive: closures, higher-order functions, call/apply/bind
- Error Handling: try-catch-finally, custom errors
- Advanced DOM: dynamic elements, attributes, forms
- ES6+ Advanced: symbols, sets, maps, optional chaining, nullish coalescing

3. Asynchronous JavaScript

- Event Loop & Concurrency: call stack, task & microtask queue
- Callbacks & callback hell
- Promises: states, chaining, Promise.all/race/allSettled/any
- Async/Await: syntax, try-catch handling
- Timers: setTimeout, setInterval
- Fetch API: GET, POST, JSON, error handling, CORS
- Web APIs: localStorage, sessionStorage, Geolocation, Clipboard

4. Advanced JavaScript

- Object-Oriented JS: constructors, prototypes, classes, inheritance, static/private methods
- Functional Programming: pure functions, immutability, composition, currying
- Advanced Array/Object Patterns: shallow vs deep copy
- Advanced DOM: shadow DOM, slots, custom elements
- Modules & Bundling: CommonJS vs ES Modules, Webpack, Vite, Rollup

5. Expert-Level JavaScript

- JS Engine Internals: V8, JIT, hidden classes, garbage collection
- Performance Optimization: event delegation, debouncing, throttling, lazy loading
- Advanced Async Patterns: generators, async generators, observables, streams
- Design Patterns: singleton, factory, observer, strategy, MVC, Flux, Redux-like
- Metaprogramming: Proxy, Reflect, Symbols
- TypeScript: annotations, interfaces, generics, utility types
- Testing: unit (Jest, Mocha), E2E (Cypress, Playwright), TDD

- Security: XSS, CSRF, CSP, sanitization
- Advanced Browser APIs: Service Workers, WebSockets, WebRTC, IndexedDB
- Concurrency: Web Workers, SharedArrayBuffer, Atomics

6. Ecosystem & Real-World Applications

- Frameworks: React, Angular, Vue
- State Management: Redux, MobX, Zustand, Pinia
- Server-side JS: Node.js, Express, NestJS
- Tooling: Babel, ESLint, Prettier
- DevOps: CI/CD pipelines, Docker, Cloud (AWS, Firebase)
- Architecture: monorepos, micro-frontends

7. Senior-Level & Interview Prep

- Big O complexity of JS operations
- Large-scale architecture & event-driven design
- Build mini React-like framework
- Implement Redux from scratch
- Write a bundler (mini Webpack)
- Scalable Node.js microservices