JavaScript Study Guide

Keep this outline in sync with the folder structure. Each numbered section should map to a directory (or subdirectory) in the workspace, so you can grow each topic incrementally.

0. Getting Started

- Install Node.js, use browser DevTools console, run scripts with `node` and `<script>`
- Editor setup: extensions, linters, formatters, file watchers
- Enable `"use strict"`, document code with comments and JSDoc
- Basic debugging: breakpoints, `console`, performance panel

1. Language Foundations (`1.Basic/`)

- `Variables.js`: `let`/`const` vs `var`, scope rules, naming conventions
- `DataTypes.js`: primitives vs objects, `typeof`, truthy/falsy, `BigInt`, `Symbol`
- `TypeConversionAndCoercion.js`: implicit vs explicit casting, parsing numbers, handling `NaN` and `Infinity`
- `Operators.js`: arithmetic, comparison, logical, nullish coalescing, optional chaining, bitwise intro
- Add coverage for expressions vs statements, template literals, Math helpers, comments & whitespace rules

2. Control Flow (`2.Control Flow/`)

- `Conditionals/`: `if/else`, `switch`, ternary, short-circuit guards
- `Loops/`: `for`, `while`, `do...while`, `for...of`, `for...in`, iterators
- `Break and Continue/`: breaking loops, guard clauses, labeled statements
- Add invalid-state handling strategies (throwing errors or returning early)

3. Functions (`3.Functions/`)

- Declarations vs expressions, arrow functions, default/rest params
- Hoisting, scope, closures, IIFE, recursion
- `this` binding, `call`/`apply`/`bind`, higher-order patterns, composition, currying

4. Arrays & Collections (`4.Array/`)

- Creation, indexing, mutation vs non-mutating patterns
- Iteration helpers: `map`, `filter`, `reduce`, `some`, `every`, `find`
- Sorting, flattening, destructuring, spread/rest
- Include `Set`, `Map`, `WeakSet`, `WeakMap`, Typed Arrays overview

5. Objects & Prototypes (`5.Object/`)

- Object literals, property descriptors, computed keys
- Shallow vs deep copy (`Object.assign`, structuredClone, JSON approaches)
- Prototypes, inheritance chain, constructor functions, ES6 classes
- `this` behavior, getters/setters, enums, JSON serialization

6. Strings, Numbers, Dates & Regex (`6.Strings/`)

- String methods, Unicode awareness, template literals, tagged templates
- Number utilities, `Intl.NumberFormat`, precision pitfalls
- `Date` API, time zones, `Intl.DateTimeFormat`, timestamps
- Regular expressions fundamentals and common patterns

7. DOM & BOM (`7.DOM & BOM/`)

- DOM tree, selectors, traversal, manipulation, templating

- Events: bubbling vs capturing, delegation, forms, custom events
- BOM APIs: `window`, `location`, `history`, dialogs, clipboard
- Storage: `localStorage`, `sessionStorage`, cookies, `indexedDB` basics

8. Asynchronous JavaScript & Networking (`8.Asynchronous JS/`)

- Callbacks, timers, microtask vs macrotask queue, event loop diagrams
- `Promise/`: states, chaining, combinators (`all`, `race`, `allSettled`)
- `async`/`await`, error handling patterns, `try/catch`
- Fetch API, AbortController, streaming, WebSockets overview

9. Error Handling & Debugging (`9.Error Handling/`)

- `try/catch/finally`, throwing custom errors, extending `Error`
- Handling promise rejections, centralized error handlers
- Debugging workflows, stack traces, logging strategies, sourcemaps

10. Modern JavaScript Features (`10.Modern JavaScript/`)

- Modules: `import`/`export`, default vs named exports, dynamic imports
- Enhanced object literals, destructuring, rest/spread
- Classes, inheritance, static/private fields, decorators preview
- Symbols, iterators, generators, `Proxy`, `Reflect`
- Optional chaining, nullish coalescing, logical assignment operators

11. Modules & Tooling (`11.Modules & Tooling/`)

- ES modules vs CommonJS, bundlers (Vite, Webpack, Rollup), Babel basics
- npm/yarn/pnpm workflows, `package.json`, scripts, dependency management
- Transpilation, polyfills, environment variables, `.env` usage
- Build optimization concepts: tree shaking, code splitting

12. Testing & Quality (`12.Testing & Quality/`)

- Unit testing with Jest/Vitest, DOM testing libraries, snapshot considerations
- Test structure, mocks, spies, coverage thresholds
- Static analysis: ESLint, Prettier, TypeScript/tsc or JSDoc type checking
- Continuous integration basics and lint/test automation

13. Architecture & Patterns (`13.Architecture & Patterns/`)

- Module pattern, revealing module, factory vs constructor, observer
- Functional vs OOP approaches, immutable patterns, composition over inheritance
- Performance tuning: profiling, debouncing, throttling, memoization
- Security and accessibility fundamentals (XSS, CSRF, ARIA)

14. Resources & Practice (`14.Resources/`)

- Reference links: MDN, ECMAScript spec, JavaScript.info
- Coding challenge platforms, project ideas by difficulty
- Glossary of common terms, interview preparation resources
- Personal notes, experiment logs, study checklist

15. Practical Exercises (`15.Practical Exercises/`)

- `01.Warmups/` through `06.Strings/`: starter questions in `practice.js` files
- `07.DOM/`: browser playground with DOM challenges
- `08.Asynchronous/`: promise and async/await drills
- Use these folders to revisit fundamentals with hands-on coding prompts