

## Swadhin Nayak [Instagram](#)

### Complete javascript topics

| Topic Number | Topic                    | Subtopics   | Links   |
|--------------|--------------------------|---|---|
| 1            | Variables and Data Types | Understanding `var`, `let`, and `const`. Primitive data types: string, number, boolean, null, undefined. Complex data types: object, array, function. | <a href="#">link</a>  |
| 2            | Operators                | Arithmetic operators (+, -, *, /). Assignment operators (=, +=, -=, *=, /=). Comparison operators (==, ===, !=, !==). Logical operators (&&,   , !).  | <a href="#">link 1</a>  |
| 3            | Control Structures       | Conditionals: if, else if, else. Loops: for, while, do-while, for...in, for...of. Switch statements.  | <a href="#">link 1</a>  |
| 4            | Functions                | Function declaration vs. function expression. Arrow functions. Higher-order functions. Closures.  | <a href="#">link1</a><br><a href="#">link2</a><br><a href="#">link 3</a><br><a href="#">link4</a> |

|    |   |   |  |
|----|---|---|--|
| 5  | Arrays and Array Methods                      | Creating arrays. Accessing and modifying array elements. Common array methods: <code>`push()`, `pop()`, `shift()`, `unshift()`, `splice()`, `slice()`, `map()`, `filter()`, `reduce()`</code> . | <a href="#">link</a>   |
| 6  | Objects and Object-Oriented Programming (OOP) | Creating objects. Object properties and methods. Prototypes and prototypal inheritance. ES6 Classes.  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a> |
| 7  | Error Handling                                | Try...catch statements. Error objects.  | <a href="#">link1</a><br><a href="#">link 2</a><br><a href="#">link 3</a>  |
| 8  | Asynchronous JavaScript                       | Callbacks. Promises. Async/await.   | <a href="#">link1</a><br><a href="#">link 2</a><br><a href="#">link 3</a>  |
| 9  | Scope and Hoisting                            | Global scope vs. local scope. Variable hoisting.  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a> |
| 10 | ES6+ Features                                 | Template literals. Destructuring assignment. Spread and rest operators. Default parameters. Modules.  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a> |
| 11 | DOM Manipulation                              | Selecting DOM elements. Modifying DOM elements. Event handling.   | <a href="#">link 1</a><br><a href="#">link 2</a>                           |

|    |                                 |  |  |
|----|---------------------------------|--|--|
| 12 | Regular Expressions             | Creating and using regular expressions. Common metacharacters and modifiers. | <a href="#">link 1</a><br><a href="#">link 2</a>   |
| 13 | Browser APIs                    | Local Storage. Fetch API. Geolocation API. WebSockets.                       | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 6</a><br><a href="#">link 7</a><br><a href="#">Link 8</a> |
| 14 | ESLint and Code Quality         | Setting up ESLint for code linting. Enforcing code quality standards.        | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a>   |
| 15 | Functional Programming Concepts | Immutability. Pure functions. Function composition.                          | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a><br><a href="#">link 6</a>                           |

|    |                          |   |  |
|----|--------------------------|---|--|
| 16 | Design Patterns          | Module pattern. Singleton pattern. Observer pattern. Factory pattern. | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a><br><a href="#">link 6</a> |
| 17 | Testing                  | Unit testing with frameworks like Jest or Mocha. Integration testing. | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a><br><a href="#">link 6</a> |
| 18 | Debugging                | Using browser developer tools. Console debugging techniques.          | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a>   |
| 19 | Performance Optimization | Minification and concatenation. Debouncing and throttling.            | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a>   |

|    |                                       |   |  |
|----|---------------------------------------|---|--|
| 20 | Security Best Practices               | Cross-site scripting (XSS) prevention. Content Security Policy (CSP). Secure storage practices. | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a><br><a href="#">link 6</a><br><a href="#">link 7</a><br><a href="#">link 8</a> |
| 21 | Functional Reactive Programming (FRP) | Reactive programming concepts. Libraries like RxJS or Bacon.js.                                 | <a href="#">link 1</a><br><a href="#">link 2</a>   |
| 22 | TypeScript                            | Adding static typing to JavaScript. Interfaces, generics, decorators.                           | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a>   |
| 23 | Package Management (npm or yarn)      | Installing and managing packages. Package.json file.  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a>   |

|    |                                |  |   |
|----|--------------------------------|--|---|
| 24 | Concurrency Models             | Understanding event loop.<br>Concurrency models in JavaScript (single-threaded, non-blocking I/O). Event-driven programming. | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a><br><a href="#">link 6</a><br><a href="#">link 7</a><br><a href="#">link 8</a><br><a href="#">link 9</a><br><a href="#">link 10</a> |
| 25 | Memory Management Optimization | Memory profiling tools.<br>Strategies for reducing memory usage. Efficient data structures.                                  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a>  |
| 26 | Functional Programming advance | Function Currying and partial application.,call apply , bind, polyfills  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a>  |
| 27 | Metaprogramming                | Reflection and introspection.<br>Manipulating objects and functions at runtime. Proxy and Reflect API.                       | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a>  |

|    |                                     |   |  |
|----|-------------------------------------|---|--|
| 28 | Advanced Asynchronous Patterns      | Generators and iterators. Reactive programming with Observables. Async iterators and generators.                                    | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a>                           |
| 29 | Advanced Error Handling             | Custom error classes. Error boundaries in React. Handling asynchronous errors.  | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a>                           |
| 30 | Performance Optimization Techniques | Memoization. Lazy evaluation. Web Workers for parallel computing. Server-side rendering (SSR) for React applications.               | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a> |
| 31 | JavaScript Engines and Optimization | Understanding V8 engine (used in Chrome and Node.js). Just-In-Time (JIT) compilation. Optimization techniques for faster execution. | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a><br><a href="#">link 4</a><br><a href="#">link 5</a> |
| 32 | Static Analysis and Type Systems    | Flow for type checking in JavaScript. TypeScript as a statically typed superset of JavaScript. ESLint plugins for static analysis.  | <a href="#">link 1</a>   |

|    |                                       |  |  |
|----|---------------------------------------|--|--|
| 33 | Internationalization and Localization | Supporting multiple languages and locales. Date and number formatting for different regions. Translation management tools. | <a href="#">link 1</a><br><a href="#">link 2</a>                           |
| 34 | Real-Time Web Development             | WebSockets for real-time communication. Implementing chat applications, live updates, and collaborative editing.           | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a> |
| 35 | Accessibility                         | Accessibility in javascript, aria labels   | <a href="#">link 1</a><br><a href="#">link 2</a><br><a href="#">link 3</a> |
| 36 | Webpack usage with javascript         | Bundling , minification , configuration  | <a href="#">link 1</a><br><a href="#">link 2</a>                           |