## PHASE 1 – JavaScript Core Foundations *(2–4 weeks)*

Before touching frameworks or backend, you must master the fundamentals.

### 1. JavaScript Basics

- **Variables**: let, const, and var (differences & scope)
- **Data Types:** string, number, boolean, null, undefined, object, symbol, bigint
- **Operators:** +, -, *, /, %, **, comparison operators, logical operators
- **Type Conversion:** Number(), String(), Boolean()

### 2. Control Flow

- if/else, switch
- Loops: for, while, do...while, for...of, for...in
- Ternary operator: condition ? expression_if_true : expression_if_false;

### 3. Functions

- Function declarations & expressions
- Arrow functions () => {}
- Parameters & default values
- Returning values

### 4. Objects & Arrays

- Object creation, property access (dot vs bracket notation)
- **Array methods:** map, filter, reduce, forEach, find, some, every, sort

### 5. DOM Basics (Front-End Side)

- document.querySelector() / getElementById()
- Changing HTML content & styles
- Adding & removing elements
- Event listeners (click, input, etc.)

## PHASE 2 – Intermediate JavaScript *(3–6 weeks)*

This phase makes you capable of handling real-world logic.

### 6. Advanced Functions

- Callback functions
- Higher-order functions
- Closures

### 7. Asynchronous JavaScript

- setTimeout, setInterval
- Promises
- async/await
- Fetch API (making HTTP requests)
- Error handling (try...catch)

### 8. ES6+ Features

- Template literals
- Destructuring
- Spread/rest operators
- Modules (import/export)
- Optional chaining ?.

### 9. DOM & Browser APIs

- Form handling
- LocalStorage & SessionStorage
- Geolocation API
- Intersection Observer (for animations on scroll)

## PHASE 3 – Front-End Development with JavaScript *(4–8 weeks)*

Time to make interactive UI and learn a framework.

### 10. Advanced DOM & UI

- Creating dynamic lists, carousels, and modals
- Event delegation
- Handling media queries with JS
- Animations with JS & CSS

### 11. JavaScript + CSS

- Adding/removing classes for styling
- Triggering animations programmatically

### 12. Front-End Frameworks *(Pick one to start)*

- **React (most popular) – Components, Props, State, Hooks, Routing OR Vue.js – Easier learning curve**
- Learn component-based architecture
- API integration with Axios/Fetch

## PHASE 4 – Back-End JavaScript (Node.js) *(6–8 weeks)*

### 13. Node.js Basics

- What is Node.js & npm
- Reading/writing files
- Creating a server with http module
- Using require & import

### 14. Express.js Framework

- Setting up a basic server
- Routing (GET, POST, etc.)
- Middleware
- Serving static files
- Handling JSON requests

### 15. Working with Databases

- MongoDB (NoSQL) + Mongoose
- OR PostgreSQL/MySQL (SQL) + Sequelize/Prisma
- CRUD operations (Create, Read, Update, Delete)

### 16. Authentication & Security

- JWT authentication
- Password hashing with bcrypt
- Helmet.js for security headers
- CORS

## PHASE 5 – Full-Stack JavaScript *(6–10 weeks)*

### 17. Connecting Front-End & Back-End

- Build APIs in Express.js
- Fetch data from your API in React/Vue
- Handle form submissions to the backend

### 18. Deployment

- Deploy frontend (Netlify, Vercel)
- Deploy backend (Render, Railway, AWS, Heroku)
- Environment variables & production setup

### 19. Real Projects

- To-do app with authentication
- E-commerce site with cart & checkout
- Blog with CRUD functionality
- Chat app with WebSockets

**PHASE 6 – Advanced Topics *(Ongoing)***

- **TypeScript (optional but highly recommended)**
- **WebSockets for real-time apps**
- **Performance optimization**
- **Testing (Jest, Mocha)**
- **GraphQL APIs**

---

Suggested Timeline
Beginner (Core JS): 2–4 weeks
Intermediate JS: 3–6 weeks
Frontend Framework: 4–8 weeks
Backend with Node.js: 6–8 weeks
Full-Stack Projects: 6–10 weeks