1. **Variables and Data Types**: Understanding different types of variables and data types like numbers, strings, booleans, etc.
2. **Operators**: Arithmetic, comparison, logical, assignment, etc.
3. **Control Flow**: if statements, else if, else, switch, loops (for, while, do-while).
4. **Functions**: Declaring functions, parameters, return statements.
5. **Arrays**: Creating arrays, accessing elements, array methods (push, pop, shift, unshift, etc.).
6. **Objects**: Object literals, accessing object properties, methods, constructors.
7. **Scope and Hoisting**: Understanding how scope works in JavaScript, hoisting behavior.
8. **DOM Manipulation**: Basic interaction with the Document Object Model (DOM) - selecting elements, changing content, styles, etc.
9. **Events**: Handling events such as click, mouseover, submit, etc.
10. **Error Handling**: Try-catch statements for handling errors.

**Intermediate Topics:**

1. **Closures**: Understanding closures and their practical uses.
2. **Prototypes and Inheritance**: Prototypal inheritance, extending objects using prototypes.
3. **Asynchronous JavaScript**: Callbacks, Promises, async/await.
4. **AJAX**: Making asynchronous requests to the server.
5. **JSON**: Working with JSON data, parsing, stringifying.
6. **Modules**: Organizing code using modules (CommonJS, ES6 modules).
7. **ES6 Features**: Arrow functions, destructuring, template literals, let/const, etc.
8. **Functional Programming**: Higher-order functions, map, filter, reduce.
9. **Regular Expressions**: Pattern matching using regular expressions.
10. **Browser Storage**: Local Storage, Session Storage, Cookies.

**Advanced Topics:**

1. **Web APIs**: Working with various Web APIs like Geolocation, Canvas, Web Workers, etc.
2. **ES6+ Features**: Generators, Proxy, Reflect, Symbols, Sets, Maps.
3. **Performance Optimization**: Improving JavaScript performance, minimizing rendering and loading times.
4. **Design Patterns**: Common design patterns in JavaScript (Singleton, Factory, Observer, etc.).
5. **Testing**: Unit testing with frameworks like Jest, Mocha, Jasmine.
6. **Security**: Understanding common security threats in JavaScript and how to mitigate them.
7. **TypeScript**: Understanding TypeScript and its benefits.
8. **Server-Side JavaScript**: Using JavaScript on the server-side with Node.js.
9. **GraphQL**: Understanding and implementing GraphQL APIs.
10. **Advanced Frameworks/Libraries**: React, Angular, Vue.js, Express.js, etc