Complete JavaScript Topics and Practice Questions

Core JavaScript Fundamentals

1. JavaScript Introduction

- What is JavaScript?
- History and Evolution of JavaScript
- Features of JavaScript
- Advantages and Disadvantages of JavaScript
- How does JavaScript work?
- Basic JavaScript syntax
- Structure of a JavaScript program

2. JavaScript Fundamentals

- JavaScript Comments
- JavaScript Keywords and Reserved Words
- Data Types in JavaScript
- Value Types and Reference Types
- JavaScript Variables
- Types of Variables in JavaScript
- Variable Declaration (var, let, const)
- Variable Scope
- Block Scope
- Variable Hoisting
- Key Difference between Var, Let, and Const
- Primitive Types

- Implicit, Explicit, Nominal, Structuring and Duck Typing

3. JavaScript Operators

- Assignment Operators
- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Conditional (Ternary) Operators
- Bitwise Operators
- Unary Operators
- TypeOf Operator
- Operator Precedence
- Increment and Decrement Operators

4. JavaScript Control Structures

- Conditional Statements
- If statement
- If-else statement
- Switch statement
- Loop statements
- While Loop
- Do-while Loop
- For loop
- Nested for loops
- For-in loop
- For-of loop
- Break statement

- Continue statement

5. JavaScript Functions and Scopes

- Function Declarations
- Function Expressions
- Anonymous Functions
- Arrow Functions
- Function Parameters and Arguments
- Default Parameters **(V IMP)**
- Rest Parameters **(V IMP)**
- Function Return Values
- Nested Functions
- Recursion
- Function Scope
- Global Scope
- Block Scope
- Lexical Scoping
- Closures
- Callbacks
- Higher-Order Functions
- IIFE (Immediately Invoked Function Expressions)
- this, call, apply and bind

6. JavaScript Objects

- Introduction to Objects
- Object Literals
- Object Properties
- Object Methods

- Creating Objects
- Using new Object()
- Constructor Functions
- Factory Functions
- Object.create()
- Accessing Object Properties (Dot notation, Bracket notation)
- Adding New Properties
- Updating Property Values
- Deleting Properties
- Checking Property Existence (hasOwnProperty(), in operator)
- Looping through Object Properties
- Object.keys()
- Object.values()
- Object.entries()
- Object Destructuring **(V IMP)**

JavaScript Built-in Objects and Methods

7. Array Object

- Creating Arrays
- Array Properties (length, constructor)
- Array Methods:
    - push(), pop(), shift(), unshift()
    - splice(), slice()
    - concat(), join()
    - indexOf(), lastIndexOf(), includes()
    - find(), findIndex()
    - map(), forEach(), filter()

    - reduce(), reduceRight()
    - some(), every()
    - sort(), reverse()
    - fill(), copyWithin() ?
    - flat(), flatMap()
- Array Destructuring **(V IMP)**
- Spread Operator with Arrays **(V IMP)**

8. String Object

- String Properties (length, constructor)
- String Methods:
    - charAt(), charCodeAt()
    - toUpperCase(), toLowerCase()
    - slice(), substring(), substr()
    - indexOf(), lastIndexOf(), includes()
    - replace(), replaceAll()
    - trim(), trimStart(), trimEnd()
    - split()
    - repeat()
    - startsWith(), endsWith()
    - padStart(), padEnd()
- Template Literals
- String Interpolation

10. Date Object

- Creating Dates
- Date Methods:

- getDate(), getMonth(), getFullYear(), getDay()
- setDate(), setMonth(), setFullYear()
- toDateString(), toLocaleDateString()
- getTime(), Date.now()

11. Number Object

- Number Properties (MAX_VALUE, MIN_VALUE, NaN, Infinity)
- Number Methods:
  - toFixed(), toPrecision()
  - Number(), parseInt(), parseFloat()
  - isNaN(), isFinite()

12. ES6+ Features

- Let and Const
- Arrow Functions
- Template Literals
- Destructuring Assignment **(V IMP)**
- Default Parameters **(V IMP)**
- Rest and Spread Operators **(V IMP)**
- Enhanced Object Literals
- Classes
- Modules (import/export)
- Symbols
- Iterators and Generators
- Map and Set
- WeakMap and WeakSet
- Promises
- Async/Await

13. Asynchronous JavaScript

- Callbacks
- Callback Hell
- Promises
- Promise.then(), Promise.catch(), Promise.finally()
- Promise.all(), Promise.race(), Promise.allSettled()
- Async/Await
- Fetch API
- Handling API Errors

14. Object-Oriented Programming (OOP)

- Classes and Objects
- Constructor Functions
- Class Constructor
- this Keyword
- this, call, apply and bind
- Prototypes and Inheritance
- Prototype Chain
- Object.prototype
- Prototype Inheritance
- Class Inheritance (extends)
- Super keyword
- Static Methods
- Getters and Setters
- Private and Public Fields
- Encapsulation
- Abstraction

- Polymorphism

15. Error Handling

- Try-catch-finally
- Throwing Exceptions
- Error Object
- Custom Error Classes
- Async Error Handling

16. Regular Expressions (RegEx)

- Creating Regular Expressions
- RegEx Methods (match(), test(), replace(), search())
- RegEx Flags (g, i, m)
- Common Patterns
- Character Classes
- Quantifiers
- Anchors

21. JavaScript Engines and Performance (https://dev.to/jeetvora331/difference-between-microtask-and-macrotask-queue-in-the-event-loop-4i4i)

- Call Stack
- Event Loop
- Message Queue
- Microtasks vs Macrotasks
- JavaScript Engines (V8, SpiderMonkey, etc.)
- Memory Management
- Garbage Collection
- setTimeout, setInterval

22. Design Patterns (https://www.patterns.dev/vanilla/singleton-pattern/) (VIMP)

- Module Pattern
- Singleton Pattern
- Factory Pattern
- Observer Pattern
- Constructor Pattern
- Prototype Pattern
- Decorator Pattern

23. Functional Programming

- Pure Functions (https://medium.com/@supraja_miryala/pure-functions-and-side-effects-in-javascript-409973f46c87)
- Side Effects
- Immutability
- Higher-Order Functions
- Function Composition
- Currying
- Partial Application
- Map, Filter, Reduce

24. Modules and Imports

- ES6 Modules
- Import and Export
- Default Exports
- Named Exports
- Dynamic Imports

- CommonJS vs ES6 Modules

JavaScript Practice Questions

Variables and Data Types

1. Declare variables using var, let, and const and explain the differences
2. Check the data type of different variables
3. Convert string to number and number to string
4. Demonstrate type coercion with examples
5. Show the difference between null and undefined

Operators

1. Create a calculator using arithmetic operators
2. Compare two values using comparison operators
3. Use logical operators to combine conditions
4. Implement conditional (ternary) operator
5. Show operator precedence with examples

Control Structures

1. Write if-else statements for different conditions
2. Create a switch statement for multiple cases
3. Use different types of loops (for, while, do-while)
4. Implement nested loops
5. Use break and continue statements

Functions

1. Create function declarations and expressions
2. Write arrow functions
3. Implement functions with default parameters

4. Use rest parameters in functions
5. Create recursive functions (factorial, fibonacci)
6. Demonstrate closures
7. Write higher-order functions
8. Create callback functions
9. Implement IIFE

Objects and Arrays

1. Create and manipulate objects
2. Use different methods to create objects
3. Access object properties using dot and bracket notation
4. Loop through object properties
5. Create and manipulate arrays
6. Use array methods (map, filter, reduce, etc.)
7. Implement array destructuring
8. Sort arrays of objects

Asynchronous Programming

1. Create and use Promises
2. Handle Promise rejections
3. Use Promise.all() and Promise.race()
4. Convert callbacks to Promises
5. Implement async/await
6. Fetch data from APIs
7. Handle API errors

ES6+ Features

1. Use template literals

2. Implement destructuring for arrays and objects
3. Use spread and rest operators
4. Create classes and inheritance
5. Use modules (import/export)
6. Work with Map and Set
7. Create generators and iterators

Object-Oriented Programming

1. Create classes with constructors
2. Implement inheritance using extends
3. Use getters and setters
4. Create static methods
5. Implement private fields
6. Demonstrate polymorphism
7. Use composition over inheritance

Design Patterns

1. Implement Module pattern
2. Create Singleton pattern
3. Use Factory pattern
4. Implement Observer pattern
5. Create Constructor pattern

Functional Programming

1. Write pure functions
2. Avoid side effects
3. Implement function composition
4. Use currying

5. Create partial applications
6. Use immutability principles

Error Handling and Testing

1. Use try-catch-finally blocks
2. Create custom error classes
3. Handle async errors
4. Write unit tests
5. Mock functions and objects

Algorithms and Data Structures

1. Implement sorting algorithms (bubble, merge, quick)
2. Create search algorithms (linear, binary)
3. Implement stack and queue
4. Create linked list operations
5. Build binary trees
6. Implement hash tables

Problem Solving

1. FizzBuzz implementation
2. Palindrome checker
3. Anagram detector
4. Prime number generator
5. Factorial calculation
6. Fibonacci sequence
7. String reversal
8. Array rotation
9. Find missing number

10. Two sum problem

https://github.com/shrutikapoor08/Learn-Web-Development-Checklist?tab=readme-ov-file

https://javascript.info/javascript-specials