Frontend Developer Interview Topics - JavaScript (2024)

# 1. Core JavaScript Concepts

## 1.1 Closures

- Creating closures

- Lexical scope

- Practical use cases (e.g., private variables)

## 1.2 Hoisting

- Variable hoisting (var, let, const)

- Function hoisting

- Hoisting behavior in block scope vs function scope

## 1.3 this Keyword and Context

- this in global scope

- this in object methods

- this in event handlers

- Arrow functions vs traditional functions and this

## 1.4 Prototypes and Inheritance

- Prototype chain

- Object inheritance using prototypes

- ES6 class inheritance and comparison to prototypes

# 2. Array Methods (Functional Programming)

- map() - Transforming arrays, Using callback functions

- filter() - Filtering arrays based on condition

- reduce() - Accumulating values, Reducing arrays to a single value (e.g., sum, product)

- find() - Finding the first matching element

- forEach() - Iterating over arrays without returning values

- some() & every() - Checking conditions in arrays (any or all)

# 3. Event Handling & DOM Manipulation

## 3.1 Event Delegation

- Delegating events to parent elements

## 3.2 Event Propagation

- Capturing and bubbling phases

- stopPropagation() and preventDefault()

## 3.3 DOM Manipulation

- Selecting elements (querySelector, getElementById)

- Modifying elements (attributes, styles, textContent)

- Removing elements

# 4. Algorithmic Thinking (Data Structures & Algorithms)

- Searching Algorithms - Linear search, Binary search

- Sorting Algorithms - Bubble sort, insertion sort, QuickSort, merge sort

- Recursion - Recursive functions for complex problems (e.g., factorial, Fibonacci)

- Dynamic Programming - Memoization, tabulation techniques

# 5. Promises and Asynchronous JavaScript

## 5.1 Promises

- Creating and resolving promises

- Promise.all(), Promise.race()

## 5.2 async/await

- Syntax and usage

- Error handling with try-catch

## 5.3 Fetch API

- Making HTTP requests

- Handling responses and errors

# 6. ES6+ (Modern JavaScript Features)

## 6.1 Arrow Functions

- Differences with traditional functions (syntax, this binding)

- Implicit return

## 6.2 Destructuring

- Array destructuring

- Object destructuring

- Nested destructuring

## 6.3 Spread and Rest Operators

- Spread in arrays and objects

- Rest in function parameters

## 6.4 Template Literals

- String interpolation

- Multi-line strings

- Tagged template literals

# 7. Object-Oriented Programming (OOP)

## 7.1 Classes and Inheritance

- Creating classes using ES6 class syntax

- Constructor functions

- Inheritance via extends and super

## 7.2 this in Classes

- Method binding

- Handling this in constructors and methods

## 7.3 Factory Functions

- Creating objects without using classes

# 8. Higher-Order Functions

- Currying - Transforming functions to accept one argument at a time

- Debouncing - Limiting the rate of function calls for better performance (e.g., input fields, search filters)

- Throttling - Restricting the execution of functions over time (e.g., scroll events)

# 9. Error Handling

- Try-Catch - Handling errors in synchronous and asynchronous code

- Custom Errors - Creating and throwing custom error types

# 10. Modules (ES6 Modules)

- import/export - Default exports vs named exports, Importing modules in ES6+

# 11. Browser APIs & Performance Optimization

- Local Storage, Session Storage - Storing and retrieving data from browser storage

- Performance Optimizations - Debouncing and throttling for event handling, Reducing reflows and repaints in the DOM, Lazy loading, batching DOM updates

# 12. Web Components & Frameworks Knowledge

- React (optional but helpful) - React hooks (useState, useEffect), State and props management, Virtual DOM

- Vue/Angular (optional but helpful) - Key features of Vue/Angular for comparison or projects