Amdocs- Frontend Developer

Interview Process

Round 1: HR Round
Round 2: Assignment
Round 3: Coding Round
Round 4: HR Round

Interview Questions

1. You are given an array consisting of 'N' positive integers where each integer is either 0 or 1 or 2, your task is to sort the given array in non-decreasing order.
2. Take two numbers as input and swap them and print the swapped values.
3. What are the features of HTML-5?
4. What is Prototype Chaining in JS?
5. What is the Same-origin policy ?
6. What are media elements ?
7. Explain promises and its 3 states .
8. Explain the Lifecycle of Components in React.
9. Explain the CSS Box Model .
10. What are callbacks?
11. Explain the difference between "==" and "==="?
12. What is JavaScript, and what role does it play in web development?
13. How do you declare variables in JavaScript?
14. Can you explain the concept of data types in JavaScript?
15. What are functions in JavaScript, and how do you define and invoke them?
16. How do you prioritize tasks when faced with multiple deadlines?
17. Describe a situation where you had to work as part of a team to achieve a goal.
18. How do you handle errors and ensure data integrity when making API requests?
19. Can you explain the concept of Cross-Site Scripting (XSS) and how to prevent it?
20. What are some common security vulnerabilities in web applications, and how do you mitigate them?

Solutions

Q1: You are given an array consisting of 'N' positive integers where each integer is either 0 or 1 or 2. Your task is to sort the given array in non-decreasing order.
A1: To sort an array consisting of only 0s, 1s, and 2s, you can use the Dutch National Flag algorithm by Dijkstra, which operates in O(N) time. You maintain three pointers (low, mid, and

high) and rearrange the array by swapping elements based on their value, ensuring that all 0s are moved to the beginning, all 2s to the end, and 1s remain in the middle.

Q2: Take two numbers as input and swap them and print the swapped values.
A2: To swap two numbers in most programming languages, you can use a temporary variable. For example, in JavaScript:

```
let temp = a;
a = b;
b = temp;
```

After swapping, print the values of `a` and `b` to see the swapped results.

Q3: What are the features of HTML-5?
A3: HTML-5 introduced several new features, including semantic elements like `<header>`, `<footer>`, and `<article>`, support for multimedia elements like `<audio>` and `<video>`, local storage, new form input types, the `<canvas>` element for 2D drawing, and improved accessibility and performance.

Q4: What is Prototype Chaining in JS?
A4: Prototype chaining in JavaScript is a mechanism by which objects inherit properties and methods from other objects. When an object's property or method is accessed, the JavaScript engine looks for it in the object itself and then continues searching up the prototype chain until it finds the desired property/method or reaches the end of the chain (null).

Q5: What is the Same-origin policy?
A5: The Same-origin policy is a security measure implemented in web browsers that restricts scripts on one origin from interacting with resources on a different origin. This prevents malicious websites from accessing sensitive data on another website via scripts, ensuring data integrity and user privacy.

Q6: What are media elements?
A6: Media elements in HTML-5 refer to the `<audio>` and `<video>` elements that allow for the embedding of audio and video content in web pages without requiring external plugins. These

elements support native playback controls and can be easily integrated with JavaScript for custom functionalities.

Q7: Explain promises and its 3 states.
A7: Promises in JavaScript represent the eventual outcome of an asynchronous operation. A promise has three states: **Pending** (the operation is ongoing), **Fulfilled** (the operation completed successfully), and **Rejected** (the operation failed). Promises enable better handling of asynchronous code by avoiding callback hell.

Q8: Explain the Lifecycle of Components in React.
A8: React components have a lifecycle that includes three main phases: **Mounting** (when the component is first rendered), **Updating** (when the component's state or props change), and **Unmounting** (when the component is removed from the DOM). Each phase has specific lifecycle methods (in class components) like `componentDidMount`, `componentDidUpdate`, and `componentWillUnmount` to manage component behavior.

Q9: Explain the CSS Box Model.
A9: The CSS Box Model describes the rectangular boxes generated for elements in a web page, including content, padding, border, and margin. The content area is the actual content, padding surrounds the content, the border encloses the padding, and the margin is the space outside the border that separates the element from others.

Q10: What are callbacks?
A10: Callbacks are functions passed as arguments to other functions that are executed after the completion of an asynchronous operation. They are widely used in JavaScript to handle events and asynchronous tasks like API requests and file reading, ensuring the code runs in the desired sequence.

Q11: Explain the difference between "==" and "==="?
A11: In JavaScript, `==` is the equality operator that checks for value equality after performing type coercion, while `===` is the strict equality operator that checks for both value and type equality, without type conversion. It's generally recommended to use `===` to avoid unexpected results from type coercion.

Q12: What is JavaScript, and what role does it play in web development?
A12: JavaScript is a high-level, interpreted programming language that plays a critical role in web development by enabling dynamic and interactive features on websites. It is used to manipulate the DOM, handle events, validate forms, and communicate with servers, making web pages responsive and engaging.

Q13: How do you declare variables in JavaScript?
A13: Variables in JavaScript can be declared using `var`, `let`, or `const`. `var` is function-scoped and can be redeclared; `let` is block-scoped and cannot be redeclared in the same scope; `const` is also block-scoped but cannot be reassigned after its initial value is set.

Q14: Can you explain the concept of data types in JavaScript?
A14: JavaScript has dynamic typing, meaning variables can hold any type of data. The basic data types include **Number**, **String**, **Boolean**, **Undefined**, **Null**, **Symbol**, and **BigInt**. There are also complex data types like **Object**, which includes arrays, functions, and custom objects.

Q15: What are functions in JavaScript, and how do you define and invoke them?
A15: Functions in JavaScript are blocks of reusable code designed to perform a specific task. They can be defined using the `function` keyword or as arrow functions. Functions are invoked by calling their name followed by parentheses `()`, optionally passing arguments to them.

Q16: How do you prioritize tasks when faced with multiple deadlines?
A16: When faced with multiple deadlines, prioritization can be done by assessing the urgency and importance of each task, setting clear goals, breaking down tasks into smaller steps, and focusing on completing high-impact tasks first. Effective time management, communication, and delegation are also key to handling multiple deadlines.

Q17: Describe a situation where you had to work as part of a team to achieve a goal.
A17: In a team project, clear communication, collaboration, and mutual support were essential in achieving our goal. We divided responsibilities based on each member's strengths, held regular meetings to track progress, and ensured that everyone contributed equally, leading to the successful completion of our objective.

Q18: How do you handle errors and ensure data integrity when making API requests?
A18: Handling errors in API requests involves using try-catch blocks, checking response statuses, validating data, and implementing retry mechanisms. To ensure data integrity, input validation, proper authentication, and encryption should be used, along with consistent logging and monitoring to catch issues early.

Q19: Can you explain the concept of Cross-Site Scripting (XSS) and how to prevent it?
A19: Cross-Site Scripting (XSS) is a security vulnerability that allows attackers to inject malicious scripts into web pages viewed by others. It can be prevented by properly escaping user input, using Content Security Policy (CSP), validating and sanitizing input on the server side, and employing frameworks that automatically prevent XSS.

Q20: What are some common security vulnerabilities in web applications, and how do you mitigate them?
A20: Common security vulnerabilities include SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and insecure direct object references. Mitigation strategies include input validation, parameterized queries, secure coding practices, implementing proper authentication and authorization, and regular security audits.