

Modern JS Common Interview Questions

These are some of the common questions asked in modern JavaScript interviews. I've thoroughly explained each concept in my lectures, so if you've gone through them carefully, these questions should be very clear to you. Additionally, all the live coding examples related to these questions are available in my lectures, so feel free to refer back to them. All the best!

JavaScript Fundamentals & Execution

1. What is the difference between the **global execution context** and the **function execution context**?
2. Explain how the **call stack** and **event loop** interact during asynchronous operations.
3. How does **hoisting** work for var, let, const, and function declarations?

Variables & Scope

4. Why is let preferred over var in most cases?
5. What will be the output of the following snippet and why?

```
console.log(x);  
let x = 5;
```

Functions & Closures

6. How does this behave in **arrow functions** vs **regular functions**?
7. Create a closure to maintain a private counter in JavaScript.
8. When would you use a **getter** and **setter** in JavaScript objects?

Objects & Memory

9. How is memory allocated for objects in JavaScript (stack vs heap)?
10. What's the difference between a **shallow copy** and a **deep copy**? Give examples using objects.

Destructuring, Optional Chaining & Nullish

11. Convert the following to use **object destructuring**:

```
const person = { name: 'Ajay', age: 30 };
```

12. When would you prefer nullish coalescing (??) over logical OR (||)?
13. What is the purpose of **optional chaining** (?.)? Give a real-world use case.

Array Methods

14. What's the difference between map() and forEach()?

Modern JS Common Interview Questions

15. Solve this: Return the sum of all even numbers from an array using `filter()` and `reduce()`.
16. What's the difference between `splice()` and `slice()`? Which one mutates the original array?

Function Manipulation & Prototypes

17. How does `call()`, `apply()`, and `bind()` differ in usage?
18. What is the prototype chain? How is it used for inheritance?
19. What is `__proto__` vs `prototype`?

Advanced Concepts

20. What are **truthy** and **falsy** values in JavaScript? Give 3 examples of each.
21. Explain the use of the **spread** operator in cloning and merging objects.
22. What's the difference between **rest** and **spread** operators?

Control Flow & Patterns

23. When would you use the **ternary operator** over `if/else`?
24. What is **short-circuit evaluation** in JavaScript? Show with `&&` and `||`.

Data Structures in JS

25. When would you use a **Map** over an Object?
26. What are the advantages of using a **Set** over an array?

Event Handling & Optimization

27. What is **event delegation** and why is it useful?
28. Explain **debouncing** and **throttling** with real-world examples.

Async Programming

29. What is the difference between a **Promise** and `async/await`?
30. Create a function that fetches data using `fetch` and handles errors using `try...catch`.