

Basic Questions

1. **What is a Promise in JavaScript?**
 - Explain its purpose and how it works.
 2. **What are the different states of a Promise?**
 - Describe `pending`, `fulfilled`, and `rejected`.
 3. **What is the difference between `Promise.all` and `Promise.allSettled`?**
 4. **How is `Promise.race` different from `Promise.any`?**
 5. **What are the advantages of using Promises over callbacks?**
-

Intermediate Questions


6. **How do you handle errors in Promises?**
 - Explain using `.catch` and `try-catch` with `async/await`.
 7. **What happens if one Promise in `Promise.all` fails?**
 8. **Explain the use case of `Promise.allSettled`.**
 9. **Can you explain how `Promise.race` works with an example?**
 10. **What is `async/await`, and how does it simplify working with Promises?**
-

Scenario-Based Questions

11. **How would you fetch data from multiple APIs simultaneously?**
 - Example: Combine `Promise.all` with `fetch`.
 12. **How do you cancel a Promise once it is initiated?**
 13. **What happens if there is no `.catch` block and a Promise is rejected?**
 14. **How would you handle the fastest response among multiple APIs using Promises?**
 - Use `Promise.race`.
-

Code-Based Questions

javascript

 Copy code


```
const delay = (ms) => new Promise((resolve) => setTimeout(resolve, ms));
delay(2000).then(() => console.log("Done after 2 seconds"));
```

Promise Interview Questions

15. Write a Promise to simulate a delay of 2 seconds before resolving.
 - 16.
 16. Implement `Promise.all` manually.
 17. Given an array of Promises, find the first successful one.
 18. Write a function to handle an array of Promises and return the first resolved value using `Promise.any`.
-

Debugging and Problem-Solving

javascript

 Copy code

```
Promise.resolve(42)
  .then((value) => {
    console.log(value);
    throw new Error("Something went wrong");
  })
  .catch((err) => console.error(err));
```

19. What will happen in the following code?
 - 20.
 - **Answer:** Logs 42 first, then logs the error message.
 20. How would you debug a situation where a Promise stays in the `pending` state forever?
-

Asynchronous Programming with Promises

21. What is the difference between synchronous and asynchronous code?
 22. Why is `async/await` preferred over `.then` and `.catch`?
 23. What happens if `await` is used outside of an `async` function?
 24. What is the difference between microtasks and macrotasks in JavaScript?
 - Examples: `Promise.then` VS `setTimeout`.
-

Real-Life Use Cases

25. How would you use Promises for retrying an API request?
26. How would you load multiple scripts in parallel using Promises?
27. How can you sequentially execute multiple asynchronous tasks with Promises