

## Question:

Think yourself as an interviewer and let's say you have been provided with 100 students to evaluate and you have been told that all the students have javascript knowledge. Now, what will be the general questions that you will ask to evaluate the students?

## Answer:

The questions that are uncommon and not asked in general are the best questions to ask.

# JavaScript Interview Questions:

---

### 1. What is Closure in JavaScript?

Closure is a function that has access to the outer function's variables.

```
function outer() {  
  let x = 10;  
  
  function inner() {  
    x++;  
    return x;  
  }  
  
  return inner;  
}  
  
let func = outer();  
console.log(func());
```

### 2. What are callbacks?

- Callbacks are functions that are passed as arguments to other functions.
- Example: Give me a call when you are done.

Examples: setTimeout, setInterval, addEventListener, methods of arrays like map, filter, reduce, etc.

### 3. What is callback hell in JavaScript?

Callback hell is a situation when you have multiple nested callbacks.

```
function execA(callback) {  
  setTimeout(() => {  
    console.log("Task A is completed!");  
    callback();  
  }, 1000);  
}
```

```
function execB(callback) {
  setTimeout(() => {
    console.log("Task B is completed!");
    callback();
  }, 1000);
}

function execC(callback) {
  setTimeout(() => {
    console.log("Task C is completed!");
    callback();
  }, 1000);
}

execA(function () {
  execB(function () {
    execC(function () {
      console.log("All tasks are completed!");
    });
  });
});
```

#### 4. What is a promise?

A promise is a object of an asynchronous operation that may be either resolved or rejected in the future.

```
const promise = new Promise((resolve, reject) => {
  // do something
  // generating random number
  const number = Math.floor(Math.random() * 10);
  if (number % 2 === 0) {
    resolve(number);
  } else {
    reject(number);
  }
});

promise
  .then((number) => {
    console.log(`Even number: ${number}`);
  })
  .catch((number) => {
    console.log(`Odd number: ${number}`);
  });
```

#### 5. What is async/await?

Async/await is a new way to write asynchronous code in JavaScript. It is a combination of promises and generators.

```
async function fetchData() {  
  const response = await  
  fetch("https://jsonplaceholder.typicode.com/posts");  
  const data = await response.json();  
  console.log(data);  
}
```

6. What is the difference between let and var?
7. What is the difference between == and ===?
8. What is the difference between null and undefined?
9. What is hoisting in JavaScript?
10. What is the event loop in JavaScript?
11. What is the difference between call, apply, and bind?
12. What is the difference between function declaration and function expression?
13. What is the difference between var, let, and const?
14. What is the difference between map, filter, and reduce?
15. What is the difference between window and document?
16. What is the difference between local storage and session storage?
17. What is the difference between cookies and local storage?
18. Explain event delegation in JavaScript?
19. Explain about higher-order functions in JavaScript?
20. What is the difference between local scope and global scope.
21. instanceof operator in JavaScript?
22. Promise Chaining
23. Pure functions
24. Debouncing and Throttling
25. What are prototypes in JavaScript?
26. What is infinite currying in JavaScript?
27. WeakMap and WeakSet
28. Temporal dead zone
29. Array buffer
30. Event capturing
31. Event bubbling
32. this keyword, class, object, constructor, prototype
33. shallow copy and deep copy
34. querySelector

## React Interview Questions:

---

1. How is Virtual DOM created in react and painting is happening behind the scenes?
2. Data flow in React?
3. What is the difference between state and props?
4. What is MVC architecture?
5. props drilling
6. what are side effects in React?

7. Class components vs Functional components.
8. redux
9. hooks
10. useEffect and limitations.
11. fetch and axios
12. lifecycle methods - class components
13. Babel
14. error boundary
15. optimization techniques in React

## NodeJS Interview Questions:

---

1. Explain about streams and buffers in NodeJS?
2. What is a middleware in NodeJS?
3. What is a fork in NodeJS?
4. What is a single thread in NodeJS?
5. filesystem in NodeJS

## Databases - MySQL & MongoDB Interview Questions:

---

1. Aggregate in mongodb
2. populate in mongodb
3. joining data in mongodb
4. DDL & DML in mysql
5. table creation in mysql
6. CRUD operations in mysql