



Published in Bits and Pieces



Ravi Sharma

[Follow](#)

Mar 30, 2022 . 3 min read . Listen

Save



# 10 Common JavaScript Interview Questions (and Answers)

## 1. Find the frequency of elements in array

**Method 1: Using Reduce method of array**

```
let arr = ["hello", "world", "java", "hello", "java"]\n\nfunction countWord(p) {\n    let result = p.reduce((allNames, name) => {\n        if (name in allNames) {\n            allNames[name]++\n        } else {\n            allNames[name] = 1\n        }\n        return allNames;\n    }, {});\n\n    return result;\n}\n\nconsole.log(countWord(arr));
```

## Method 2: Using an Object

```
let arr = ["hello", "world", "java", "hello", "java"]

function countWord(p) {
    var count = {};
    p.forEach((item) => {
        if (count[item]) {
            count[item]++;
        } else {
            count[item] = 1
        }
    })
    return count;
}

console.log(countWord(arr));
```

## 2. Group items on the basis of age of given array of object

```
● ● ●

let people = [
  { name: 'Alice', age: 21 },
  { name: 'Max', age: 20 },
  { name: 'Jane', age: 20 }
];

function groupBy(ar) {
  var check = {};

  ar.forEach((item) => {
    if (!check[item.age]) {
      check[item.age] = [item];
    } else {
      check[item.age].push(item);
    }
  })
  return check;
}

console.log(groupBy(people));

// Output
{
  '20': [ { name: 'Max', age: 20 }, { name: 'Jane', age: 20 } ],
  '21': [ { name: 'Alice', age: 21 } ]
}
```

### 3. Program to check a string with balanced brackets.

```
const isValid = (str) => {
    let s = str.replace(/\s+/g, '');
    if (s.length % 2 !== 0) return false;

    const stack = [];
    const map = new Map([
        [ '(', ')' ],
        [ '[ ', ']' ],
        [ '{ ', '}' ]
    ]);

    for (let i = 0; i < s.length; i += 1) {
        if (map.has(s[i])) {
            stack.push(map.get(s[i]));
        } else if (s[i] !== stack.pop()) {
            return false;
        }
    }
    return stack.length === 0;
};

let str1 = "({} [] ({}))";
let str2 = "})}";

console.log(isValid(str1)); // true
console.log(isValid(str2)); // false
```

#### 4. Find the pairs of array element for which sum is equal to given target value (Two Sum Problem)

```
let ar = [1, 2, 3, 4, 6, 7, 8, 9];

function twoSum(ar, target) {
    let hash = {};
    let sum = []
    ar.forEach((item) => {
        let diff = target - item;

        if (hash[diff.toString()] !== undefined) {
            sum.push([item, diff]);
        }
        hash[item.toString()] = item;
    })
    return sum;
}

console.log(twoSum(ar, 9));

// Output [ [ 6, 3 ], [ 7, 2 ], [ 8, 1 ] ]
```

## 5. Find the missing number from unsorted array with O(n) complexity

### Algorithm

1. Create a variable `sum = 1` which will store the missing number and a counter variable `c = 2`.
2. Traverse the array from start to end.
3. Update the value of sum as `sum = sum - array[i] + c` and update `c` as `c++`.
4. Print the missing number as a sum.

A screenshot of a mobile browser displaying a code challenge. The page has a dark purple header with three dots at the top. Below the header, there is some placeholder text: "Copy & Paste your code here". The main content area contains the following JavaScript code:

```
let ar = [2, 7, 8, 5, 1, 4, 3, 6];

function missing(a) {
    let l = a.length;
    let sum = 1;

    for (let i = 0; i < l - 1; i++) {
        sum += a[i];
    }

    return sum;
}

console.log(missing(ar));
```

[Open in app ↗](#)[Sign up](#)[Sign In](#)

A screenshot of a mobile browser displaying the continuation of the code challenge. The page has a dark purple header with three dots at the top. Below the header, there is some placeholder text: "Copy & Paste your code here". The main content area continues the previous code:

```
return sum;
}

console.log(missing(ar));
```

## 6. Find the missing number from sorted array with O(n) complexity

```
let ar = [1, 3, 4, 6, 7, 8, 10];

function missing(a) {
    let missing = [];

    for (let i = 0; i < ar.length; i++) {
        if (!(ar[i + 1] - ar[i] == 1) && !(ar[i + 1] == undefined)) {
            missing.push(ar[i] + 1);
        }
    }
    return missing;
}

console.log(missing(ar));
// Output [ 2, 5, 9 ]
```

471 | 5

## 7. Find the nth largest element in a sorted array

```
let arr = [12,34,21,14,67,35,64,25];
arr.sort((a, b) => a - b);

const nthLargest = arr[arr.length - n] // if n= 2

console.log(nthLargest);
// Output 64
```

## 8. Remove duplicates from an array and return unique values in O(n) complexity.

```
let arr = [1, 2, 3, 4, 3, 6, 5, 4];

function unique(arr) {
    let items = {}
    arr.forEach((item) => {
        if (!items[item]) {
            items[item] = item;
        }
    });
    return Object.values(items);
}
console.log(unique(arr));
// Output [1,2,3,4,6,5]
```

## 9. Print all duplicate elements of an array

```
let arr = [1, 1, 5, 6, 7, 7, 8, 9, 3, 4, 4];

function removeDuplicates(ar) {
    let result = ar.filter((item, index) => {
        return ar.indexOf(item) !== index;
    });
    return result;
}

console.log(removeDuplicates(arr));
// Output: [1,7,4]
```

## 10. Collect books from array of objects and return collection of books as an array

```
let friends = [ {  
    name: 'Anna',  
    books: ['Bible', 'Harry Potter'],  
    age: 21  
}, {  
    name: 'Bob',  
    books: ['War and peace', 'Romeo and Juliet'],  
    age: 26  
}, {  
    name: 'Alice',  
    books: ['The Lord of the Rings', 'The Shining'],  
    age: 18  
}]  
  
let result = friends.reduce((pre, curr) => {  
    return [...pre, ...curr.books]  
}, []);  
  
console.log(result);  
  
// Output:  
[  
  'Bible',  
  'Harry Potter',  
  'War and peace',  
  'Romeo and Juliet',  
  'The Lord of the Rings',  
  'The Shining'  
]
```

I hope you have found this quick list of common JavaScript interview questions and answers useful. Thank you for reading!