



Coding Questions for Objects

js



Nikhil | FullStack Dev
@codewithnikhill



>>>

1. Check if an Object is Empty



```
function isEmpty(obj) {  
    return Object.keys(obj).length === 0;  
}  
  
console.log(isEmpty({})); // Output: true  
console.log(isEmpty({a: 1})); // Output: false
```

>>>

2. Merge Two Objects



```
function mergeObjects(obj1, obj2) {  
    return { ...obj1, ...obj2 };  
}  
  
console.log(mergeObjects({a: 1, b: 2}, {b: 3, c: 4}));  
// Output: {a: 1, b: 3, c: 4}
```

>>>

3. Clone an Object (Shallow Copy)

```
● ● ●

function cloneObject(obj) {
    return { ...obj };
}

const original = {a: 1, b: 2};
const copy = cloneObject(original);
console.log(copy); // Output: {a: 1, b: 2}
console.log(copy === original); // Output: false
```

>>>

4. Deep Clone an Object (Deep Copy)

```
● ● ●

function deepClone(obj) {
    return JSON.parse(JSON.stringify(obj));
}

const original = {a: 1, b: {c: 2}};
const copy = deepClone(original);

console.log(copy); // Output: {a: 1, b: {c: 2}}
console.log(copy.b === original.b); // Output: false
```

>>>

5. Count Properties in an Object



```
function countProperties(obj) {  
    return Object.keys(obj).length;  
}  
  
console.log(countProperties({a: 1, b: 2, c: 3}));  
// Output: 3
```

>>>

6. Get Keys and Values as Separate Arrays

```
const person = {  
    name: 'Alice',  
    age: 25,  
    city: 'New York'  
};  
  
// Get keys  
const keys = Object.keys(person);  
  
// Get values  
const values = Object.values(person);  
  
console.log(keys); // Output: ['name', 'age', 'city']  
console.log(values); // Output: ['Alice', 25, 'New York']
```

>>>

7. Check if an Object Has a Property

```
● ● ●

function hasProperty(obj, key) {
    return obj.hasOwnProperty(key);
}

console.log(hasProperty({a: 1, b: 2}, 'a')); // Output: true

console.log(hasProperty({a: 1, b: 2}, 'c')); // Output: false
```

>>>

8. Convert Object to Array of [Key, Value] Pairs



```
function objectToPairs(obj) {  
    return Object.entries(obj);  
}  
  
console.log(objectToPairs({a: 1, b: 2, c: 3}));  
// Output: [['a', 1], ['b', 2], ['c', 3]]
```

>>>

9. Get All Methods of an Object

```
● ● ●

function getMethods(obj) {
  return Object.keys(obj).filter(key => typeof obj[key] === 'function');
}

const obj = {
  a: 1,
  b: 2,
  foo() { return 'foo'; },
  bar() { return 'bar'; }
};

console.log(getMethods(obj)); // Output: ['foo', 'bar']
```