

# JavaScript Quiz Questions and Answers 2



## JavaScript Code Examples and Questions



Laurence Svekis [BaseScripts.com](https://basescripts.com)

What is a closure in JavaScript and how do you create one?	2
What is the difference between == and === in JavaScript?	3
How do you declare a variable in JavaScript?	3
What is the difference between null and undefined in JavaScript?	3
How do you check if a variable is an array in JavaScript?	4
What is the difference between var and let in JavaScript?	4
How do you create a new object in JavaScript?	5
How do you create a new object in JavaScript?	5
How do you add an element to the end of an array in JavaScript?	6
<code>console.log(arr); // logs [1, 2, 3, 4]</code>	6
How do you loop through an object in JavaScript?	6
How do you check if a variable is defined in JavaScript?	7
How do you remove an element from the beginning of an array in JavaScript?	7
What is the difference between let and const in JavaScript?	7
How do you convert a string to a number in JavaScript?	8
How do you concatenate two arrays in JavaScript?	8
What is the difference between null and undefined in JavaScript?	9
How do you reverse the order of elements in an array in JavaScript?	9
How do you check if a variable is an object in JavaScript?	9
How do you convert a number to a string in JavaScript?	10
How do you check if a string contains a substring in JavaScript?	10
How do you loop through an object in JavaScript?	10
How do you get the current date and time in JavaScript?	11

How do you convert a string to uppercase or lowercase in JavaScript?	11
How do you create a new array with unique values in JavaScript?	12
How do you find the maximum or minimum value in an array in JavaScript?	12
How do you create a new array with elements that meet a certain condition in JavaScript?	12

## What is a closure in JavaScript and how do you create one?

A closure is a function that has access to variables in its outer (enclosing) function's scope chain. This allows the closure to retain access to those variables even after the outer function has returned. Closures are created by defining a function inside another function and returning it. Here is an example:

```
function outerFunction() {  
  let outerVariable = "Hello";  
  
  function innerFunction() {  
    console.log(outerVariable);  
  }  
  
  return innerFunction;  
}
```

```
let closure = outerFunction(); // closure now refers to  
innerFunction  
closure(); // logs "Hello" because innerFunction has  
access to outerVariable
```

## What is the difference between == and === in JavaScript?

The == operator compares values for equality after performing type coercion, while the === operator compares values for equality without performing type coercion. For example:

```
1 == "1" // true because "1" is coerced to a number
1 === "1" // false because "1" is not equal to 1
without coercion
```

## How do you declare a variable in JavaScript?

You can declare a variable in JavaScript using the var, let, or const keywords. var and let are used for declaring variables that can be reassigned, while const is used for declaring variables that cannot be reassigned. For example:

```
var x = 5; // var keyword
let y = 10; // let keyword
const z = 15; // const keyword
```

## What is the difference between null and undefined in JavaScript?

null is a value that represents the intentional absence of any object value, while undefined is a value that represents an uninitialized or nonexistent value. For example:

```
let x = null; // x is intentionally set to null
let y; // y is undefined because it has not been
initialized
```

## How do you check if a variable is an array in JavaScript?

You can check if a variable is an array in JavaScript using the `Array.isArray()` method. For example:

```
let x = [1, 2, 3];
let y = "hello";
console.log(Array.isArray(x)); // true
console.log(Array.isArray(y)); // false
```

## What is the difference between var and let in JavaScript?

The main difference between `var` and `let` in JavaScript is that `var` has function scope, while `let` has block scope. This means that variables declared with `var` are accessible within the entire function in which they are declared, while variables declared with `let` are only accessible within the block in which they are declared. For example:

```
function example() {
  var x = 5;
  if (true) {
    var x = 10;
  }
  console.log(x); // logs 10 because x is reassigned
  // within the same function scope
}
```

```
function example2() {  
  let y = 5;  
  if (true) {  
    let y = 10;  
  }  
  console.log(y); // logs 5 because y is only  
  accessible within the block scope of the if statement  
}
```

## How do you create a new object in JavaScript?

You can create a new object in JavaScript using either object literal notation `{}`, constructor notation `new Object()`, or class notation `class MyClass {}`. For example:

```
let obj1 = {}; // object literal notation  
let obj2 = new Object();
```

## How do you create a new object in JavaScript?

You can create a new object in JavaScript using either object literal notation `{}`, constructor notation `new Object()`, or class notation `class MyClass {}`. For example:

```
let obj1 = {}; // object literal notation  
let obj2 = new Object(); // constructor notation  
class MyClass {  
  constructor(x, y) {  
    this.x = x;
```

```
    this.y = y;
  }
}
let obj3 = new MyClass(5, 10); // class notation
```

## How do you add an element to the end of an array in JavaScript?

You can add an element to the end of an array in JavaScript using the `push()` method. For example:

```
let arr = [1, 2, 3];
arr.push(4);
```

```
console.log(arr); // logs [1, 2, 3, 4]
```

## How do you loop through an object in JavaScript?

You can loop through an object in JavaScript using a `for...in` loop. For example:

```
let obj = {x: 1, y: 2, z: 3};
for (let prop in obj) {
  console.log(`${prop}: ${obj[prop]}`);
}
// logs "x: 1", "y: 2", "z: 3"
```

## How do you check if a variable is defined in JavaScript?

You can check if a variable is defined in JavaScript using the `typeof` operator or the `undefined` keyword. For example:

```
let x;  
console.log(typeof x === "undefined"); // true because  
x has not been initialized  
console.log(x === undefined); // true because x has not  
been initialized
```

## How do you remove an element from the beginning of an array in JavaScript?

You can remove an element from the beginning of an array in JavaScript using the `shift()` method. For example:

```
let arr = [1, 2, 3];  
arr.shift();  
console.log(arr); // logs [2, 3]
```

## What is the difference between `let` and `const` in JavaScript?

The main difference between `let` and `const` in JavaScript is that `let` allows you to reassign the variable to a new value, while `const` does not. For example:

```
let x = 5;  
x = 10; // reassigns x to a new value
```

```
const y = 15;  
y = 20; // throws a TypeError because y is a constant  
and cannot be reassigned
```

## How do you convert a string to a number in JavaScript?

You can convert a string to a number in JavaScript using the `Number()` function or the `parseInt()` or `parseFloat()` methods. For example:

```
let str = "123";  
let num1 = Number(str);  
let num2 = parseInt(str);  
let num3 = parseFloat(str);  
console.log(num1, num2, num3); // logs 123 123 123
```

## How do you concatenate two arrays in JavaScript?

You can concatenate two arrays in JavaScript using the `concat()` method or the spread operator (`...`). For example:

```
let arr1 = [1, 2, 3];  
let arr2 = [4, 5, 6];  
let arr3 = arr1.concat(arr2);  
let arr4 = [...arr1, ...arr2];  
console.log(arr3); // logs [1, 2, 3, 4, 5, 6]  
console.log(arr4); // logs [1, 2, 3, 4, 5, 6]
```



## What is the difference between null and undefined in JavaScript?

null is a value that represents the intentional absence of any object value, while undefined is a value that represents an uninitialized or non-existent value. For example:

```
let x = null; // x is intentionally set to null
let y; // y is undefined because it has not been
        initialized
```

## How do you reverse the order of elements in an array in JavaScript?

You can reverse the order of elements in an array in JavaScript using the reverse() method. For example:

```
let arr = [1, 2, 3];
arr.reverse();
console.log(arr); // logs [3, 2, 1]
```

## How do you check if a variable is an object in JavaScript?

You can check if a variable is an object in JavaScript using the typeof operator or the instanceof operator. For example:

```
let obj = {x: 1, y: 2};
console.log(typeof obj === "object"); // true
console.log(obj instanceof Object); // true
```

## How do you convert a number to a string in JavaScript?

You can convert a number to a string in JavaScript using the `toString()` method or by concatenating the number with an empty string (`""`). For example:

```
let num = 123;
let str1 = num.toString();
let str2 = num + "";
console.log(str1, str2); // logs "123" "123"
```

## How do you check if a string contains a substring in JavaScript?

You can check if a string contains a substring in JavaScript using the `includes()` method or the `indexOf()` method. For example:

```
let str = "Hello, world!";
console.log(str.includes("world")); // true
console.log(str.indexOf("world") !== -1); // true
```

## How do you loop through an object in JavaScript?

You can loop through an object in JavaScript using a `for...in` loop or the `Object.keys()` method. For example:

```
let obj = {x: 1, y: 2, z: 3};
for (let key in obj) {
  console.log(key, obj[key]);
}
```

```
}  
// logs "x 1", "y 2", "z 3"  
  
let keys = Object.keys(obj);  
for (let i = 0; i < keys.length; i++) {  
  let key = keys[i];  
  console.log(key, obj[key]);  
}  
// logs "x 1", "y 2", "z 3"
```

## How do you get the current date and time in JavaScript?

You can get the current date and time in JavaScript using the `Date()` constructor. For example:

```
let now = new Date();  
console.log(now); // logs the current date and time
```

## How do you convert a string to uppercase or lowercase in JavaScript?

You can convert a string to uppercase or lowercase in JavaScript using the `toUpperCase()` method or the `toLowerCase()` method. For example:

```
let str = "Hello, world!";  
let uppercase = str.toUpperCase();  
let lowercase = str.toLowerCase();  
console.log(uppercase, lowercase); // logs "HELLO,  
WORLD!" "hello, world!"
```

## How do you create a new array with unique values in JavaScript?

You can create a new array with unique values in JavaScript using the `Set()` constructor and the spread operator (`...`). For example:

```
let arr = [1, 2, 2, 3, 3, 3];
let uniqueArr = [...new Set(arr)];
console.log(uniqueArr); // logs [1, 2, 3]
```

## How do you find the maximum or minimum value in an array in JavaScript?

You can find the maximum or minimum value in an array in JavaScript using the `Math.max()` function or the `Math.min()` function, along with the spread operator (`...`). For example:

```
let arr = [1, 2, 3];
let max = Math.max(...arr);
let min = Math.min(...arr);
console.log(max, min); // logs 3 1
```

## How do you create a new array with elements that meet a certain condition in JavaScript?

You can create a new array with elements that meet a certain condition in JavaScript using the `filter()` method. For example:

```
let arr = [1, 2, 3, 4, 5];
let evenArr = arr.filter(num => num % 2 === 0);
```

```
console.log(evenArr); // logs [2, 4]
```