

JavaScript Interview Questions

1

What do you understand about JavaScript?

- JavaScript is a popular web scripting language and is used for client-side and server-side development
- The JavaScript code can be inserted into HTML pages that can be understood and executed by web browsers while also supporting object-oriented programming abilities



What's the difference between JavaScript and Java?



JavaScript	Java
JavaScript is an object-oriented scripting language.	Java is an object-oriented programming language.
JavaScript applications are meant to run inside a web browser.	Java applications are generally made for use in Operating Systems and Virtual Machines.
JavaScript does not need compilation before running the application code.	Java source code needs a compiler before it can be ready to run in realtime.

3

What are the various data types in JavaScript?

- JavaScript has many data types to provide basic functionality needed for a web application
- They are mentioned in this table

Data type	Description
Boolean	For true and false values
Null	For empty or unknown values
Undefined	For variables that are only declared and not defined or initialized
Number	For integer and floating-point values
String	For characters and alphanumeric values
Object	For collections or complex values
Symbols	For unique identifiers or objects

4

What are the features of JavaScript?

- Lightweight, interpreted programming language
- Cross-platform compatible
- Open-source
- Object-oriented
- Integration with other backend as well as frontend technologies
- Used especially for the development of network-based applications



5

What are the advantages of JavaScript?

- **Enhanced interaction** - JavaScript adds interaction to otherwise static web pages and makes them react to users' inputs
- **Quick feedback** - There is no need for a web page to reload when running JavaScript
- **Rich User Interface** - JavaScript helps in making the UI of web applications look and feel much better
- **Frameworks** - JavaScript has countless frameworks and libraries that are extensively used for developing web applications and games of all kinds



6

How do you create an object in JavaScript?

Since JavaScript is essentially an object-oriented scripting language, it supports the usage of objects while developing web applications

```
const student = {  
    name: 'John',  
    age: 17  
}
```

7

How do you create an array in JavaScript?

Here is a very simple way of creating arrays in JavaScript using the array literal:

```
var a = [];  
var b = ['a', 'b', 'c', 'd', 'e'];
```

8

What are some of the built-in methods in JavaScript?

- JavaScript has many built-in methods to provide functionality needed for a web app
- Some of them are mentioned in this table

Method	Description
Date()	Returns the present date and time
Concat()	Joins two strings and returns the new string
Push()	Adds an item to an array
Pop()	Removes and also returns the last element of an array
Round()	Rounds off the value to the nearest integer and returns it
Length()	Returns the length of a string

9

What are the scopes of a variable in JavaScript?

The scope of a variable basically implies where the variable has been declared or defined in a JavaScript program

There are two scopes of a variable:

- **Global Scope** - Global variables, having global scope are available everywhere in a JavaScript code
- **Local Scope** - Local variables are accessible only within a function in which they are defined

**10**

What is 'this' keyword in JavaScript?

- The 'this' keyword in JavaScript refers to the currently calling object
- It is commonly used in constructors to assign values to object properties

**11**

What are the conventions of naming a variable?

- Variable names cannot be similar to that of reserved keywords. For example, var, let, const, etc.

`var var;`

11

What are the conventions of naming a variable?

- Variable names cannot be similar to that of reserved keywords. For example, var, let, const, etc.
- Variable names cannot begin with a numeric value; they must only begin with a letter or an underscore character

var 123; 

var a123; 

12

What is Callback in JavaScript?

- A Callback is a JavaScript function that is passed to another function as an argument or a parameter

A higher order function takes a function as a parameter
↓
const higherOrderFunction = (callback) → {
 return callback()
}
↑
A callback is a function that is passed as an argument

12

What is Callback in JavaScript?

- A Callback is a JavaScript function that is passed to another function as an argument or a parameter
- This function is to be executed whenever the function that it is passed to gets executed

A higher order function takes a function as a parameter
↓
const higherOrderFunction = (callback) → {
 return callback()
}
↑
A callback is a function that is passed as an argument

13

How do you debug a JavaScript code?

- All modern web browsers like Chrome, Firefox, etc. have an inbuilt debugger that can be accessed anytime by pressing the relevant key, usually the F12 key



13

How do you debug a JavaScript code?

- All modern web browsers like Chrome, Firefox, etc. have an inbuilt debugger that can be accessed anytime by pressing the relevant key, usually the F12 key
- We can also debug a JavaScript code inside a code editor that we use to develop a JavaScript application—for example, Visual Studio Code, Atom, Sublime Text, etc.



14

What's the difference between function declaration and function expression?

```
function abc() {  
    return 5;  
}
```

```
var a = function abc() {  
    return 5;  
}
```

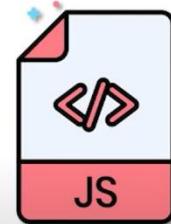
Function declaration	Function expression
Declared as a separate statement within the main JavaScript code	Created inside an expression or some other construct
Can be called before the function is defined	Created when the execution point reaches it; can be used only after that
Offers better code readability and better code organization	Used when there is a need for a conditional declaration of a function

15

What are the ways of adding JavaScript code in an HTML file?

There are majorly two ways of embedding JavaScript code:

- We can write JavaScript code within the script tag in the same HTML file
- We can import a JavaScript source file into an HTML document



16

What do you understand about cookies?

- A cookie is generally a small data that is sent from a website and stored on the user's machine by a web browser that was used to access the website



16

What do you understand about cookies?

- A cookie is generally a small data that is sent from a website and stored on the user's machine by a web browser that was used to access the website
- Cookies are used to remember information for later use and also to record the browsing activity on a website



17

How would you create a cookie?

The simplest way of creating a cookie using JavaScript is as below:

```
document.cookie = "key1 = value1; key2 = value2; expires = date";
```

18

How would you read a cookie?

We can simply use the `document.cookie` string that contains the cookies in the form of key, value pair

```
key1 = value1;
```

19

How would you delete a cookie?

To delete a cookie, we can just set an expiration date and time; specifying the correct path of the cookie is a good practice

```
function delete_cookie(name) {  
    document.cookie = name + "=; Path=/; Expires=Thu,  
    01 Jan 1970 00:00:01 GMT;";  
}
```

20

What is the difference between let and var?

- Both let and var are used for variable and method declarations in JavaScript
- So there isn't much of a difference between these two besides that while var keyword is scoped by function, let keyword is scoped by a block



21

What are Closures in JavaScript?

- Closures provide a better, and concise way of writing JavaScript code for the developers and programmers
- Closures are created whenever a variable that is defined outside the current scope is accessed within the current scope

```
function hello(name) {  
    var message = "Hello " + name;  
    return function hello() {  
        console.log(message);  
    };  
}  
//generate closure  
var helloWorld = hello("World");  
//use closure  
helloWorld();
```

22

What are the arrow functions in JavaScript?

Arrow functions are a short and concise way of writing functions in JavaScript

```
const helloWorld = () => {  
    console.log("Hello World!");  
};
```

23

What are the different ways an HTML element can be accessed in a JavaScript code?

Accessor	Description
getElementByClass('classname')	Gets all the HTML elements that have the specified classname
getElementById('idname')	Gets an HTML element by its ID name
getElementsByTagName('tagname')	Gets all the HTML elements that have the specified tagname
querySelector()	Takes CSS style selector and returns the first selected HTML element

24

What are the ways of defining a variable?

There are three ways of defining a variable in JavaScript:

Var - This is used to declare a variable and the value can be changed at a later time within the JavaScript code

Const - We can also use this to declare/define a variable but the value is constant throughout the JavaScript program and cannot be modified

Let - This mostly implies that the values can be changed at a later time within the JavaScript code

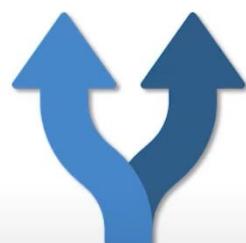
```
let abc = 'Hello';
```

25

What are imports and exports in JavaScript?

simplilearn

- Imports and exports help in writing modular code for our JavaScript applications
- With the help of imports and exports, we can split a JavaScript code into multiple files in a project



25

What are imports and exports in JavaScript?

```
export const sqrt = Math.sqrt;
export function square(x) {
    return x * x;
}
export function diag(x, y) {
    return sqrt(square(x) + square(y));
}
```

This file exports two functions that calculate the squares and diagonal of the input respectively

```
import { square, diag } from "calc";
console.log(square(4)); // 16
console.log(diag(4, 3)); // 5
```

Here we import those functions and pass input to those functions to calculate square and diagonal

26

What is the difference between document and window in JavaScript?



Document

The document comes under the windows object and can also be considered as its property

Window

Window in JavaScript is a global object that holds the structure like variables, functions, location, etc.

27

What are some of the JavaScript Frameworks and their uses?

JavaScript has a collection of many frameworks that aim towards fulfilling the different aspects of the web application development process

Some of the prominent frameworks are:

- React - Frontend development of a web application
- Angular - Frontend development of a web application
- Node - Backend or server-side development of a web application



29

What is the difference between undefined and null in JavaScript?

Undefined	Null
Undefined means a variable has been declared but a value has not yet been assigned to that variable.	Null is actually an assignment value that we can assign to any variable that is meant to contain no value.

30

What is the difference between Session storage and Local storage in JavaScript?

Session storage	Local storage
The data stored in session storage gets expired or deleted when a page session ends	Websites store some data in local machine to reduce loading time; this data does not get deleted at the end of a browsing session

31

How do you empty an array in JavaScript?

There are a few ways in which we can empty an array in JavaScript:

- By assigning array length to 0

```
var arr = [1, 2, 3, 4];
arr.length = 0;
```

31

How do you empty an array in JavaScript?

There are a few ways in which we can empty an array in JavaScript:

- By assigning array length to 0
- By assigning an empty array
- By popping the elements of the array
- By using the splice array function

```
var arr = [1, 2, 3, 4];
arr.splice(0, arr.length);
```

32

What is the difference between Event Capturing and Event Bubbling?

Event Capturing

This process starts with capturing the event of the outermost element and then propagating it to the innermost element

Event Bubbling

This process starts with capturing the event of the innermost element and then propagating it to the outermost element

33

What is the Strict mode in JavaScript?

Strict mode in JavaScript introduces more stringent error-checking in a JavaScript code

- While in Strict mode, all variables have to be declared explicitly, values cannot be assigned to a read-only property, etc.
- We can enable strict mode by adding 'use strict' at the beginning of a JavaScript code, or within a certain segment of code

```
"use strict";
let a = 10;
let b = 20;
let c = a + b;
```

34

What would be the output of the below JavaScript code?

- The output of this JavaScript code will be 10undefined
- The if condition statement in the code evaluates using eval
- Hence, eval(function abc(){}) will return function abc(){}
Inside the if statement, executing typeof abc returns undefined because the if statement code executes at run time while the statement inside the if condition is being evaluated

```
var a = 10;
if (function abc(){})
{
a += typeof abc;
}
console.log(a);
```

35

Can you write a JavaScript code for adding new elements in a dynamic manner?

```
<script type="text/javascript">
function addNode() {
    var newP = document.createElement("p");
    var textNode = document.createTextNode("This is a new text node");
    newP.appendChild(textNode);
    document.getElementById("firstP").appendChild(newP);
}
</script>
```

36

What is the difference between Call and Apply?

Call	Apply
In the call() method, arguments are provided individually along with a 'this' value.	In the apply() method, arguments are provided in the form of an array along with a 'this' value.

37

What would be the output of the below JavaScript code?

- This is basically a simple example of object-oriented programming
- Therefore, the output will be 'abc' as we are accessing the property of the Student object.

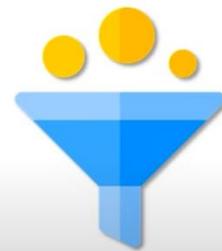
```
var Student = {  
    college: "abc",  
};  
var stud1 = Object.create(Student);  
delete stud1.college;  
console.log(stud1.college);
```

39

How do you remove duplicates from a JavaScript array?

There are two ways in which we can remove duplicates from a JavaScript array:

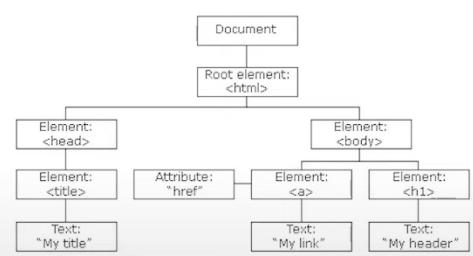
- By using the filter method - In order to call the filter() method, three arguments are required. These are namely array, current element, and index of the current element.



40

Can you draw a simple JavaScript DOM
(Document Object Model)?

Here is a simple diagram of Document Object Model (DOM) of a webpage



create

```
const ride = new Promise((resolve, reject) => {
  if (arrived) {
    resolve('driver arrived 🚗');
  } else {
    reject('driver bailed 😞');
  }
});
```

consume

```
ride
  .then(value => {
    console.log(value);
    // driver arrived 🚗
  })
  .catch(error => {
    console.log(error);
    // driver bailed 😞
  })
```

catch

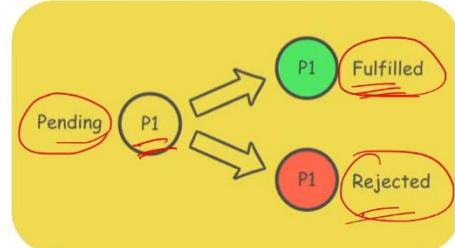
handle rejection

```
const promise = new Promise((resolve, reject) => {
  // Perform asynchronous operation for eg: setTimeout()
  // Call `resolve` function when the operation succeeds
  // Call `reject` function when the operation encounters an error
});
```



Important points about promises:

1. Promises in JavaScript are a way to handle **asynchronous operations**.
2. A Promise can be in one of three states: **Pending, resolved, or rejected.**
3. A promise represents a value that **may not be available yet** but will be available at some point in the **future**.



```
const promise = new Promise((resolve, reject) => {
  // Perform asynchronous operation for eg: setTimeout()
  // Call `resolve` function when the operation succeeds
  // Call `reject` function when the operation encounters an error
});
```



```
// Promise.all() method is used to handle multiple promises concurrently.  
const promise1 = new Promise((resolve) => setTimeout(resolve, 1000, "Hello"));  
const promise2 = new Promise((resolve) => setTimeout(resolve, 2000, "World"));  
const promise3 = new Promise((resolve) => setTimeout(resolve, 1500, "Happy"));  
  
// Promise.all takes an array of promises as input and returns a new promise.  
Promise.all([promise1, promise2, promise3])  
  .then((results) => {  
    console.log(results); // Output: ['Hello', 'World', 'Happy']  
  })  
  .catch((error) => {  
    console.error("Error:", error);  
  });
```

```
// callback = a function that is passed as an argument  
//           to another function.
```

```
//           used to handle asynchronous operations:  
//           1. Reading a file  
//           2. Network requests  
//           3. Interacting with databases
```

```
//           "Hey, when you're done, call this next."
```