JavaScript Interview Questions

edureka!

1. What is the significance of using JS?

JavaScript, an object scripting language used in web pages along with HTML. An HTML is a markup language that can create static pages and without JavaScript, it would still be static. JavaScript can add dynamic nature to a website by making it more interactive. It makes website navigation easier so that designers can guide the visitors with additional information or guide them through walkthroughs. Visual effects can also be achieved at a much better level with JavaScript as it can create special effects like rollover for images.

2. How can we check if a given value is a number or not in JS?

There are two ways to check whether the given value is a number or not.

- By using **isNaN().** It is a global variable that is assigned to the window object in the browser. If it returns false, the value **is** a number.

- By using typeof operator. It will return a string with the value 'number' if used on a number value.

3. What is the difference between null and undefined?

In JavaScript, **undefined** represents that the variable has been declared, but it has not been assigned any value yet. It means its value and type are undefined. For example:

```
var x;
console.log(x); //shows undefined.
console.log(typeof x); //shows undefined.
```

null represents that value has been assigned to a variable. It means its value is null but its type is object. For example:

```
var x = null;
console.log(x); //shows null.
console.log(typeof x); //shows object.
```

From the above examples, it should be clear that undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

4. **Difference between == vs === ?**

== operator	=== operator
It compares two variables but ignores the datatype of variables	It compares the value of two variables but also compares the datatype of those variables

for example,
var a = 1
var b = “1”
if(a==b) {
console.log(“Matched”)
}
else {
console.log(“Not Matched”)
}
Output: Matched
	for example,
	var a = 1
	var b = “1”
	if(a===b) {
	console.log(“Matched”)
	}
	else {
	console.log(“Not Matched”)
	}
	Output: Not Matched

5. Difference between setTimeout and setInterval functions in JS?

setTimeout(expression, timeout) is used to run the code/function once after the timeout.

setInterval(expression, timeout) is used to run the code/function in intervals i,e, it will run after a certain period of time.

Example:
var intervalID = setInterval(alert, 1000); // Will alert every second.
setTimeout(alert, 1000); // Will alert once, after a second.

6. Explain this keyword in JS?

This keyword can have different values depending on where it is used as:

- In a method, this keyword refers to the owner object.
- Alone, this keyword refers to the global object.
- In a function, this keyword refers to the global object.
- In a function, in strict mode, this keyword is undefined.
- In an event, this keyword refers to the element that received the event.
- Methods like call() and apply() can refer to this to any object.

7. Evaluate 3+2+"7".

57

8. Evaluate 3+"2"+7.

327

9. What is the significance of using strict in JS?

There are multiple benefits of using strict in JS. Some of them are:

- It eliminates some errors that do not display any message by changing them into throw errors.
- It fixes mistakes that can make JavaScript engines difficult to perform optimizations. Sometimes it can run faster than simple identical code.
- It prohibits some syntax likely to be defined in future versions of ECMAScript.
- It prevents or throws errors when relatively "unsafe" actions are taken (such as gaining access to the global object).
- It can disable the features that are confusing or poorly thought out.
- It makes it easier to write "secure" JavaScript.

10. What are the ways of integrating JS with HTML?

JavaScript code can be integrated with HTML document by using HTML tag <script> and will wrap the JavaScript code inside. The <script> tag can be placed inside:

- <head> section of the HTML file
- <body> section of the HTML file

11. What is the difference between let, var, and const?

var	let	const
var declarations have global / function scope	let declarations have block scope	const declarations have block scope
variables can be updated and re-declared within its scope	variables can be updated but not re-declared	variables can neither be updated nor re-declared
var variables can be declared without being initialized	let variables can be declared without being initialized	const variables must be initialized during declaration

12. Difference between the splice and slice functions in JS?

splice Method	slice Method
It returns the removed items in an array	It returns the selected element in an array as a new array object
It makes changes in the original array	It doesn't make any changes in the original array
It can take multiple arguments	It can take two arguments

13. What is a Ternary Operator?

In javascript, Ternary Operator takes three operands: a condition followed by a question mark (?), then an expression that should be executed if the condition is true (truthy) followed by a colon (:), and finally an expression that should be executed if the condition is false (falsy). This operator is considered a shortcut for the if-else statement.
<condition> ? <value> : <value>

14. Explain Arrow functions in JS?

Javascript Arrow function was introduced in the ES6 version of JavaScript. It enables you to write functions in a cleaner way as compared to regular functions. For example,

This function

```
// normal function expression
let a = function(a, b,c) {
   return a + b+c;
}
```

can be written as

```
// using arrow functions
let a = (a, b,c) => a + b+c;
```

15. Explain IIFE functions in JS?

Immediately Invoked Function Expression (IIFE) is one of the most popular design patterns in JavaScript. Functions and variables defined in this do not conflict with other functions & variables even though they have the same name.

```
(function () {
    //write your code here
})();
```

16. **Difference between Call, Apply, and Bind functions in JS?**

- The bind method is used to create a new function that sets this keyword to the specified object. It is used when you want that function should to be called later with a specific context. It accepts comma-separated arguments.

- The call method sets this keyword inside the function and executes that function immediately. It accepts a comma-separated list of arguments.

- The apply method is similar to the call method, but it accepts an array of arguments instead of comma-separated values.

17. **Differentiate between Local Storage and Session Storage?**

Local storage	Session storage
Stored data persists until it deleted explicitly	Stored data persists until the tab is opened
Changes made will be saved and reflected for all current and future visits to the site	Changes made will be saved and reflected for the current page in that tab until it is closed
For example, data will remain even after browser restart	For example, data will be lost after browser restart

18. **Explain the Hoisting concept in JavaScript?**

In Javascript, Hoisting is a default behavior where variables and function declarations are moved to the top of their scope before code execution. It considers functions and variables are moved to the top of their scope, regardless of their declaration, whether it has global or local scope.

19. **Explain ways to handle asynchronous data in JavaScript?**

There are three ways to handle asynchronous data calls in JavaScript. The are:

- Callback functions
- Promises
- Async & Await

20. **What are Callback Functions in JavaScript?**

When a function is passed as an argument to another function is known as **Callback Functions**. It is a technique that allows functions to call another function and will only run once another function has finished.

For example:

```
function Learning(x) {
  document.getElementById("demo").innerHTML = x;
}
function Display() {
  Learning("Welcome to edureka!");
}
Display();
```

Output: Welcome to edureka!

21. Explain Promises in JavaScript?

A **promise** is an object used for a value that is unknown at that time. It contains both the producing code and calls to the consuming code. Whenever it is executed, it should either call a success callback i.e., myResolve(result value), or the error callback i.e., myReject(error object). It helps associate handlers to use asynchronous actions and find out the success value or failure reason. These asynchronous methods return values that can return the single value at some point in the future instead of returning it immediately. A Promise object can have one of the three states:

- Pending: It represents that the operation is neither fulfilled nor rejected.
- Fulfilled: It represents that the operation was completed successfully.
- Rejected: It represents that the operation failed.

22. Explain Async/ Await in JavaScript?

Async/ Await is considered as the extension of promises. Async is used to make sure that a promise is returned. If it is not returned, then javascript will automatically wrap it in a promise and resolve it with its value. Await can only be used within the async block and is used to wait for the promise.

23. What are Closures in JavaScript?

A **Closure** is a feature in JavaScript where an inner function has access to the outer (enclosing) function's variables, also known as the scope chain. The closure has three scope chains:

- It has access to the variables that come under its own scope i.e., variables defined between its curly brackets.
- It has access to the variables present outside the function.
- It has access to the global variables.

24. Difference between Spread and Rest Operator in JavaScript?

Rest is a parameter that helps to pass an indefinite number of parameters to a function that can be accessed in an array. It is a collection of all remaining elements into an array.

Spread is an operator that helps us to spread the value of an array across zero or more arguments in a function or elements in an array. It works exactly opposite to the rest parameter, unlike Rest parameter it unpacks the collected elements into single elements.