

JavaScript Interview Questions

1. Is JavaScript and TScript the same?

→ JavaScript : is a Scripting language (supports script) for Web pages but it is also used in non-browser environments as well. It is a powerful, lightweight, interpreted, Scripting language with first-class functions (i.e. the language supports passing functions as arguments to other functions).

TScript : is also a scripting language, much similar to JavaScript. It is a dialect of the popular ECMA Script standard reverse-engineered by Microsoft. TScript is subsidy by Microsoft and used in one of the most popular web browsers Microsoft's Internet Explorer. TScript can also be called "Microsoft's JavaScript".

2. Justify the use of let and const in javascript?

→ Earlier in javascript, developers use the var keyword for creating variables. let & const keyword is introduced in version ES6 with the vision of creating two different types of variables in javascript one is immutable and the other is mutable. The use of let and const in javascript:

- Let

let is used for variable declaration as it comes as an improvement to the var declarations. let is a block code bounded as {}. So a variable declared in a block with the let is only available for use within that block.

- Example :-

```
let greeting = "say Hi";
```

```

let times = 4;
if (time > 3) {
  let hello = "Say Hello instead";
  console.log(hello); // "Say Hello instead"
}
console.log(hello) // hello is not defined
    
```

- const

Variables declared with the `const` keyword maintain constant values. `const` declarations share some similarities with `let` declarations.

Like `let` declarations, `const` declarations can only be accessed within the block it was declared in.

`const` cannot be updated or re-declared.

This means that the value of a variable declared with `const` remains the same within its scope. It cannot be updated or re-declared. So if we declare a variable with `const`,

- Example :-

```

const greetings = "Say Hi";
greeting = "Say Hello instead"; // error: Assignment
                                to constant variable
    
```

3. Explain the `toMul` function in JavaScript?

`MUL` means a simple multiplication of number. It is a technique in which you pass one value as an argument in a function and that function returns another function to which you pass the second value and the process goes on. Multiplies two expressions. This is the functional equivalent of the `(*)` operator.

4. List the Frameworks and Data types supported by JavaScript.

→ 20 The frameworks used by JavaScript are:

- Node.js
- Angular.js
- React
- Vue.js
- Ember.js
- Meteor
- Backbone.js

→ 21 Data types supported by JavaScript are:

- Symbol
- String
- Boolean
- Null
- Undefined
- Number
- Object

5. How you can redirect a page to another page in JavaScript?

→ There are several ways to redirect the page to another page in JavaScript. These are:

• Using location.href : It is the first approach to redirect page. In this, we can go back to access the original document.

• Using location.replace : Another approach to redirect page. In this, it is not possible to navigate back to the original document by clicking on the back button as it removes the URL of the original document.

6. What is promises?

→ A JavaScript promise object contains both the producing code and calls to the consuming code.

22 A Java

• Per

• Ful

• Re

• The pro

and re

• While a

result

when a

value

when a

error or

7. How let

Var has +
or global

• It gets h
top of if

initializ

• It can l
or re-

• It's an
declare

8. What

• Cooki

• Local

• Ses

When

• A JavaScript promise object can be:

- pending

- Fulfilled

- Rejected

• The promise object supports two properties: state and result.

• While a promise object is "pending" (working), the result is undefined.

• When a promise object is "fulfilled", the result is a value.

• When a promise object is "rejected", the result is an error object.

7. How let, const, and var are different?

→ Var → It's a let → It's a const

• Var has the function let's have the block const variable has the global scope.

• It gets hoisted to the It also got hoisted to the It also got hoisted to the top of its scope and top of its scope but of its scope but initialized undefined didn't initialize didn't initialize.

• It can be updated It can only be updated It can't be updated or re-declared It can't be re-declared re-declared.

• It's an old way to It's a new way to It's also a new way to declare a variable declare variables declare a variable introduced in ES6 which introduced in ES6.

8. What are different types of client storage?

- Cookies (old school, still useful because they are sent with server requests) = 5kb
- Local Storage (very easy to use) = SMB
- Session Storage (exactly the same, only clears when the tab is closed).

9. Difference between null and undefined in javascript.

→ Null : It is the intentional absence of the value. It is one of the primitive values of JavaScript.

Undefined : It means the value does not exist in the compiler.

10. Difference between == and ===.

→ == operator does the type conversion of the operands before comparison.

→ === operator compares the values as well as the data types of the operands.

11. Use of this keyword in javascript.

→ "This" keyword refers to an object that is executing the current piece of code. It references the object that is executing the current function. If the function being referenced is a regular function, "this" refers to the global object.

12. Difference between map() and forEach() in javascript.

→ • The map() method returns a new array, whereas the forEach() method does not return a new array.

• The map() method is used to transform the elements of an array, whereas the forEach() method is used to loop through the elements of an array.

13. What is closure.

→ A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives you access to an outer function's scope from an inner function.

14. What is Hoisting in javascript
→ Javascript "Hoisting" refers to the process whereby the interpreter appears to move the declaration of functions, variables or classes to the top of their scope, prior to execution of the code.

15. What is Destructuring in javascript
→ Destructuring means to break down a complex structure into simpler parts. With the syntax of destructuring, you can extract smaller fragments from objects and arrays. It can be used for assignments and declaration of a variable.

16. What are the features in ES6
→ The let keyword, the const keyword, Arrow Functions, the ... Operator, For/of, Map, Object Set, Objects, classes, promises, Symbol, Default parameters, Function Rest parameter, String, includes(), String.startsWith(), String.endsWith(), Array.from(), Array.keys(), Array.find(), Array.findIndex(), New Math Methods, New Number properties, New Number Methods, New Global Methods, Object entries, JavaScript Modules.

17. What is Global Variable in Javascript
→ A global variable is a variable that is declared in the global scope, in other words, a variable that is visible from all other scopes. In Javascript it is a property of the global object.

18. Explain window.onload and documentReady?
→ The window.onload method gets executed after the entire web page is loaded. This includes all

22.

the elements related with DOM like the head tag, title tag and all the other tags including the style sheets, images and videos.

The onload method is used by passing a function to it.

19. Difference between Local storage and Session storage in javascript.

SessionStorage is similar to localStorage; the difference is that while data in localStorage doesn't expire, data in sessionStorage is cleared when the page session ends. Whenever a document is loaded in a particular tab in the browser, a unique page session gets created and assigned to that particular tab.

20. What are the different ways an HTML element can be accessed in a Javascript code?

- Get HTML element by Id.
- Get HTML element by .ClassName
- Get HTML element by Name
- Get HTML element by tagName.
- Get HTML element by CSS Selector.

21. What does Super keyword do? When it is used?

The super keyword refers to superclass (parent) object. If is used to call superclass methods and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name.

23.

22. What is callback in JavaScript?

→ A callback is a function passed as an argument to another function. This technique allows a function to call another function. A callback function can run after another function has finished.

23. What are the different types of event in JavaScript?

- Event Description
- onchange An HTML element has been changed.
 - onclick the user clicks an HTML element
 - onmouseover The user moves the mouse over an HTML element.
 - onmouseout The user moves the mouse away from an HTML element.
 - onkeydown The user pushes a keyboard key
 - onload The browser has finished loading the page

24. What is pure function?

A pure function is a function (a block of code) that always returns the same result if the same arguments are passed. It does not depend on any state or data change during a program's execution. Rather, it only depends on its input arguments.

25. How does new keyword works in javascript.

→ When a function is called with the new keyword, the function will be used as a constructor. new will do the following things: Create a blank, plain JavaScript object. For convenience, let's call it newinstance.

26. Truthy and Falsey values in JavaScript.

```

→ Boolean(false); // false
Boolean(undefined); // false
Boolean(null); // false
Boolean(""); // false
Boolean(NaN); // false
Boolean(0); // false
Boolean(-0); // false
Boolean("0"); // false
Boolean(true); // true
Boolean('hi'); // true
Boolean(1); // true
Boolean(11); // true
Boolean([0]); // true
Boolean([1]); // true
Boolean({}); // true
Boolean({a: 1}); // true
    
```

27. Call, Apply and bind in JavaScript.

- **call**: is a function that helps you change the context of the invoking function. In JavaScript's terms, it helps you replace the value of this inside a function with whatever value you want.
- **Apply**: is very similar to call function. The only difference is that in apply you can pass an array as an argument list.
- **Bind**: is a function that helps you create functions that you can execute later with the new context of this that is provided.

28. Function expression vs function declaration in JavaScript.

Function Expression

1. A function declaration must have a function name.

2. Function declaration does not require a variable assignment.

3. The function in function declaration can be accessed before and after the function definition.

4. These are executed before any other code.

5. Function declarations are hoisted

6. Syntax: `function geekForGeeks(paramA, paramB){
 // Set of statements
}`

Q9. Explain Higher ordered function

→ A function which takes another function as parameter is called Higher ordered function.

A function which returns another function is called Higher order function.

30. Is javascript is Single-threaded or Multi-threaded?

→ In the context of programming, parallelism is the utilization of multiple threads in an operating system. Routines are able to run at the same time regardless of execution order.

Function declaration

A function Expression is similar to a function declaration without the function name.

Function expressions can be stored in a variable assignment.

The function in function declaration can be accessed before and after the function definition.

Function expressions load & execute only when the program interpreter reaches the line of code.

Function expressions are not hoisted.

Syntax: `var geekForGeeks =
 function(paramA, paramB){
 // Set of statements
}`

JavaScript, however, is single threaded and only one line of code can be executed at any given time.