Microsoft JavaScript Interview Questions and Answers

Here are some of the most important Microsoft JavaScript interview questions with answers to help you prepare:

1. What is the difference between == and === in JavaScript?

Answer: The == operator in JavaScript compares two values for equality, after converting both values to a common type (type coercion).Additionally the === operator compares both the values and their types, so no type conversion occurs. For example 5 == '5' will return true because the string '5' is converted to the number 5. But 5 === '5' will return false because the types (number and string) are different.

2. Explain closures in JavaScript.

Answer: A closure is a function that remembers its outer variables and can access them even when the function is executed outside their scope. Closures are used in JavaScript for data encapsulation and to create private variables. For example:

function is executed outside their scope. Closures are commonly used in JavaScript for data encapsulation and to create private variables. For example:

```
function outerFunction() {
        let counter = 0;
                return function innerFunction(){
        counter++;
                return counter;
        }
}
const incrementCounter = outerFunction();
console.log(incrementCounter()); // Output: 1

console.log(incrementCounter());  // Output: 2
```

3. What is an Immediately Invoked Function Expression (IIFE)?

Answer: An IIFE is a function that is executed immediately after it is defined. It is often used to create a local scope to avoid polluting the global namespace:

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

4. What is event delegation in JavaScript?

Answer: Event delegation is a technique where a single event handler is used to manage events for multiple elements. Instead of adding event listeners to every element, you can

add a single event listener to a parent element that handles events from its children by taking advantage of event bubbling.

5. What is JavaScript and how is it different from Java?

Answer: JavaScript is a high level dynamic language used to add interactivity to web pages. It's lightweight and interpreted so it runs directly in the browser without any compilation phase. Java is a robust object oriented language used to build standalone applications and large scale enterprise systems. JavaScript is used for front end development and Java is used for back end and server side development.

6. How does this work in JavaScript?

Answer: In JavaScript this refers to the object from which the function was called. Besides the value of this can change depending on the context:

- Global context: this refers to the global object (window in browsers).
- Function context: this refers to the object that the function is a method of.
- Constructor functions: this refers to the newly created instance.
- Event handlers: this refers to the element that received the event.

7. What are Promises in JavaScript?

Answer: A Promise is an object that represents the completion or failure of an asynchronous operation. Promises are a more powerful and flexible way to handle asynchronous operations than callbacks, no more callback hell.

8. What is the bind() method in JavaScript?

Answer: The bind() method creates a new function that when called has its this set to a specific value, along with any number of arguments that were provided as an argument to the new function when it's called. It's often used to set this in callback functions.

9. What's the difference between null and undefined?

Answer: undefined is a type that means a variable has been declared but not assigned a value. Because null is an assignment value that can be assigned to a variable to mean "no value" or "empty".

10. What is hoisting in JavaScript?

Answer: Hoisting is JavaScript's default behavior of moving declarations to the top of the current scope (either global or function scope). So variables and function declarations can be used before they are declared in the code.

11. What are arrow functions and how are they different from regular functions?

Answer: Arrow functions are a shorter syntax for writing functions in JavaScript, introduced in ES6. Besides they don't have their own this context and don't have arguments, super or new.target bindings. Also Arrow functions are best for non-method functions.

12. What is the purpose of async and await in JavaScript?

Answer: async and await are used to work with Promises better in JavaScript. async functions always return a Promise and await pauses the execution of an async function until the Promise is resolved.

13. What is a higher-order function in JavaScript?

Answer: A higher-order function is a function that takes one or more functions as an argument or returns a function as its result. Additionally Functions like map, filter and reduce are examples of higher-order functions.

14. What is the prototype chain in JavaScript?

Answer: In JavaScript, objects inherit properties and methods from their prototype. Additionally this forms a prototype chain, a series of links between objects where each object points to its prototype until it reaches null.

15. How is forEach different from map in JavaScript?

Answer: forEach is used to iterate over an array and execute a function on each element but it doesn't return a new array. map on the other hand also iterates over an array but it creates a new array with the results of calling a function on every element.

16. What are modules in JavaScript and why are they important?

Answer: Modules are a way to organize and encapsulate code into separate files or blocks, for better maintainability, reusability and scope management. Besides ES6 introduced native module support in JavaScript using the import and export keywords.

17. What is destructuring?

Answer: Destructuring is a syntax that allows you to pull values out of arrays or properties out of objects into separate variables. Besides it makes extracting values from complex data structures easier.

18. What does the spread operator (...) do?

Answer: The spread operator allows an iterable (array or string) to be expanded in places where 0 or more arguments or elements are expected. Besides it can be used to copy arrays, merge arrays or pass multiple arguments to a function.

19. What are template literals?

Answer: Template literals are string literals that allow expressions. Besides they are enclosed by backticks (``) instead of quotes and can contain placeholders denoted by ${expression} which are evaluated and included in the string.

20. What does the reduce method do?

Answer: The reduce method runs a reducer function on each element of an array and returns a single output value. The reducer function takes 4 arguments: the accumulator, the current value, the current index and the array itself.

21. What is the difference between call, apply and bind in JavaScript?

Answer:

- call: Calls a function with a specified this value and arguments provided individually.
- apply: Calls a function with a specified this value and arguments provided as an array.
- bind: Returns a new function with a specified this value and optional initial arguments, without calling the function.

22. What does the try...catch statement do?

Answer: The try...catch statement allows you to handle errors in JavaScript. Besides Code that may throw an error is placed in the try block and if an error occurs the catch block executes to handle the error.

23. How does JavaScript handle asynchronous operations?

Answer: JavaScript handles asynchronous operations using callbacks, Promises, async/await and event loops. These allow tasks to run without blocking the main thread and makes the app more responsive.

24. What does Object.freeze() do?

Answer: Object.freeze() prevents an object from being modified. Additionally It makes the object immutable by disallowing addition, removal or modification of properties.

25. How does Object.assign() work?

Answer: Object.assign() copies all enumerable own properties from one or more source objects to a target object. It returns the target object, effectively merging properties from source objects into it.