Topic 3 JavaScript

Interview Questions and Answers

Q1. 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>

Q2. Explain Arrow functions in JavaScript.

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;
```

Q3. Explain IIFE functions in JavaScript.

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 and variables even though they have the same name.

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

Q4. Difference between Call, Apply, and Bind functions in JavaScript?

1. 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 be called later with a specific context. It accepts comma-separated arguments.
2. The call method sets this keyword inside the function and executes that function immediately. It accepts a comma-separated list of arguments.
3. The apply method is similar to the call method, but it accepts an array of arguments instead of comma-separated values.

Q5. Differentiate between Local Storage and Session Storage?

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

Q6. 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 that are moved. to the top of their scope, regardless of their declaration, whether it has global or local scope.

Q7. Explain ways to handle asynchronous data in JavaScript.

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

1. Callback functions
2. Promises
3. Async & Await.

Q8. What are Callback Functions in JavaScript?

When a function is passed as an argument to another function is known as Callback Function.

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!

Q9. 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 a single value at some point in the future instead of returning it immediately.

A Promise object can have one of the three states:

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

Q10. 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.

Q11. 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:

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

Q12. Difference between Spread and Rest Operators in JavaScript?

1. **Rest**

- 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.

2. **Spread**

- 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 the Rest parameter, it unpacks the collected elements into single elements.

Q13. What is debounce in JavaScript?

Debounce is a method of programming to improve the browser's performance.

If a browser has time-consuming functions and if those functions are frequently/repeatedly getting invoked, then it will affect the browser's performance.

Debouncing is a concept that is used to ensure that the time-consuming tasks do not fire so often which helps to improve the efficiency of the webpage performance.

In other words, debounce is a way of programming that limits the rate at which a time-consuming function gets called.