1. List all ES6 features of js.

Ans:

* Let and const keyword.
* Arrow functions
* Spread operator.
* Map and set.
* Promises.
* Classes.
* Rest parameter.
* JavaScript modules.

1. What is destructuring?

Ans: Destructuring is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables. That is, we can extract data from arrays and objects and assign them to variables.

1. What is a promise?

Ans: A promise is an object that encapsulates the result of an asynchronous operation.

A promise object has a state that can be one of the following:

* Pending
* Fulfilled with a value.
* Rejected for a reason.

In the beginning, the state of a promise is pending, indicating that the asynchronous operation is in progress. Depending on the result of the asynchronous operation, the state changes to either fulfilled or rejected.

Promise constructor accepts 2 callback functions resolve() and reject().

If the asynchronous operation completes successfully, the executor will call the resolve() function to change the state of the promise from pending to fulfilled with a value.

In case of an error, the executor will call the reject() function to change the state of the promise from pending to rejected with the error reason.

1. Explain map, reduce and filter method.

Ans: The map() method takes an array and a callback function as arguments. The callback function is applied to each element of the array, and the results are returned in a new array.

The filter() method takes an array and a callback function as arguments. The callback function is applied to each element of the array, and the elements that return true are returned in a new array.

The reduce() method takes an array and a callback function as arguments. The callback function is applied to each element of the array, and the results are accumulated into a single value.

1. What is higher order functions and callback functions? Difference between them.

Ans: Higher Order Function: A function that accepts a function as an argument and/or returns a function as its value.

Callback Function: A function that's passed as an argument to another function.

1. Difference between map and set.

Ans: A map is a collection of key-value pairs. Each key is unique, and each value can be any data type. Maps are often used to store data that needs to be accessed quickly, such as a lookup table.

A set is a collection of unique elements. Each element can be any data type, and the order of the elements is not important. Sets are often used to store data that needs to be unique, such as a list of unique identifiers.

1. What is hoisting in JavaScript?

Ans: In JavaScript, hoisting refers to the built-in behavior of the language through which declarations of functions, variables, and classes are moved to the top of their scope – all before code execution. In turn, this allows us to use functions, variables, and classes before they are declared.

1. What are the differences between == and ===?

Ans: In JavaScript, the === operator is a strict equality comparison operator that matches by both value and data type. The == operator is an equality comparison operator that matches by value only.

1. What are the different ways to create an object in JavaScript?

Ans:

* Object literal
* Constructor function
* Classes
* Object.create()

1. What is the event loop in JavaScript?

Ans: The event loop is a mechanism in JavaScript that allows the execution of code to be scheduled in a non-blocking way. It is a single-threaded loop that continuously checks for events and then executes the appropriate callback functions. The event loop allows JavaScript to be responsive to user input, network events, and other asynchronous operations.

1. What is asynchronous programming?

Ans: Asynchronous programming is a technique that enables your program to start a potentially long-running task and still be able to be responsive to other events while that task runs, rather than having to wait until that task has finished. Once that task has finished, your program is presented with the result. This is achieved by using callbacks, promises, or async/await functions.

1. What is the prototype chain in JavaScript?

Ans: Every object in JavaScript has a built-in property, which is called its prototype. The prototype is itself an object, so the prototype will have its own prototype, making what's called a prototype chain. The chain ends when we reach a prototype that has null for its own prototype.

1. What is the purpose of the this keyword in JavaScript?

Ans: The this keyword in JavaScript is a powerful tool that allows you to reference the current object. This can be useful in a variety of situations, such as when you need to access the properties or methods of the current object, or when you need to pass the current object to a function.

The value of the this keyword depends on the context in which it is used. In an object method, this refers to the object that owns the method. In a function, this refers to the global object. In strict mode, this is undefined.

1. What is the use of the isNaN function?

Ans: The isNaN() function is used to check if a value is NaN (Not a Number). isNaN() returns true if a number is Not-a-Number.

1. What is call, apply and bind method?

Ans: The call method sets the this inside the function and immediately executes that function.

The apply() method is similar to call(). The difference is that the apply() method accepts an array of arguments instead of comma separated values.

The bind method creates a new function and sets the this keyword to the specified object.

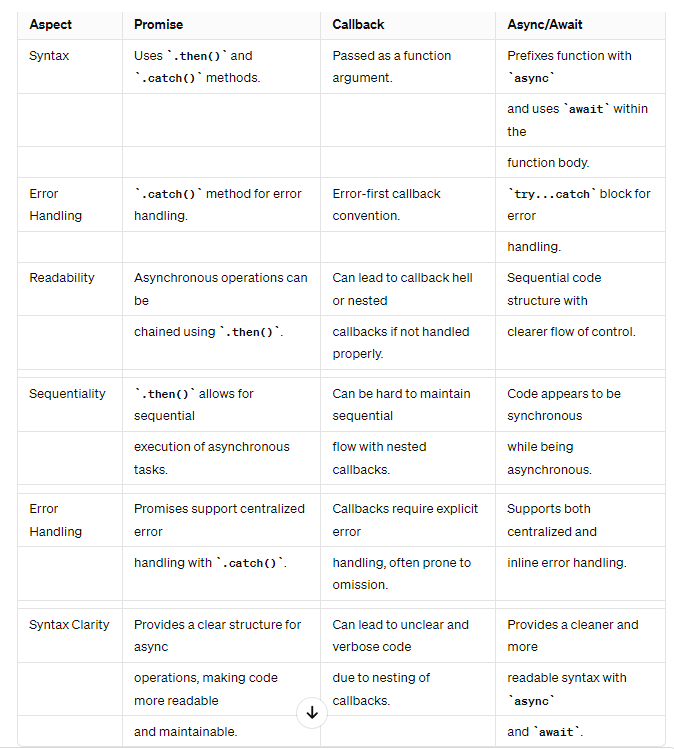
1. What is DOM?

Ans: DOM stands for Document Object Model. It's a programming interface that represents the structure of a web document. DOM represents a document as a logical tree, with each node containing an object.

1. What is callback hell?

Ans: Callback Hell is essentially nested callbacks stacked below one another forming a pyramid structure. Every callback depends/waits for the previous callback, thereby making a pyramid structure that affects the readability and maintainability of the code.

1. Difference between promise, callback and async and await.

Ans: 

1. What is IIFE and its uses?

Ans: In JavaScript, an Immediately Invoked Function Expression (IIFE) is a function that executes immediately after it is defined. IIFEs are self-contained blocks of code that are executed once. IIFEs are commonly used to create private scope in JavaScript, allowing variables and functions to be encapsulated and inaccessible from outside the function.

1. What is strict mode?

Ans: Strict mode makes it easier to write "secure" JavaScript.

Strict mode changes previously accepted "bad syntax" into real errors.

As an example, in normal JavaScript, mistyping a variable name creates a new global variable. In strict mode, this will throw an error, making it impossible to accidentally create a global variable.

1. What is web storage?

Ans: Web storage is a JavaScript API that allows websites to store data on a user's device. It's also known as DOM storage (Document Object Model storage). Web storage is an HTML5 feature that allows users to store data in key-value pairs in their browser. This allows applications to store data on the client side, so it can be accessed or manipulated later. All data stored in web storage stays in the browser and is not transferred anywhere else.

1. Difference between local storage and session storage?

Ans: A white sheet with black text

Description automatically generated

1. What is cloning? Different ways of deep cloning and shallow cloning?

Ans: Cloning in JavaScript refers to the process of creating a copy of an existing object. There are two main types of cloning: shallow cloning and deep cloning. Shallow cloning creates a new object that has the same properties as the original object, but the values of those properties are still references to the original object's values. This means that if you change the value of a property in the cloned object, the change will also be reflected in the original object.

The Object.assign() method can be used to shallow clone an object. The spread operator can also be used to shallow clone an object.

Deep cloning, on the other hand, creates a new object that has its own copies of the original object's properties. This means that if you change the value of a property in the cloned object, the change will not affect the original object. The JSON.parse() and JSON.stringify() methods can be used to deep clone an object.

1. What is self invoking functions?

Ans: Function expressions can be made "self-invoking". A self-invoking expression is invoked (started) automatically, without being called. It is also known as an Immediately Invoked Function Expression (IIFE).

1. What is closure?

Ans: 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. In JavaScript, closures are created every time a function is created, at function creation time.

1. What is function constructor?

Ans: A function constructor in JavaScript is a way to create objects using a function as a blueprint. Essentially, you define a function and then use the new keyword to instantiate new objects based on that function.

1. What is instanceof operator?

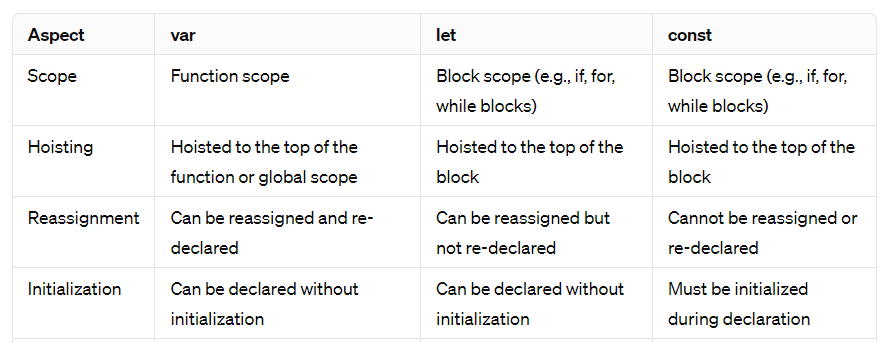
Ans: The JavaScript instanceof operator is used to check the type of an object at the run time. It returns a boolean value(true or false). If the returned value is true, then it indicates that the object is an instance of a particular class and if the returned value is false then it is not.

1. What is spread and rest operator?

Ans: The spread operator, denoted by three consecutive dots (...), is primarily used for expanding iterables like arrays into individual elements. This operator allows us to efficiently merge, copy, or pass array elements to functions without explicitly iterating through them.

While the spread operator expands elements, the rest operator condenses them into a single entity within function parameters or array destructuring. It collects remaining elements into a designated variable, facilitating flexible function definitions and array manipulation.

1. Difference between var, let and const.

Ans: 

1. What is garbage collection in js?

Ans: Garbage collection (GC) in JavaScript is a mechanism that automatically manages memory allocation and deallocation. It frees up memory that is no longer being used by the program, without the programmer having to explicitly manage it.

1. Different types of memory type.

Ans: There are two main types of memory in JavaScript:

Stack memory: This is a small, fast memory that is used to store function calls and local variables. Stack memory is allocated automatically when a function is called, and it is deallocated when the function returns.

Heap memory: This is a large, slower memory that is used to store objects and arrays. Heap memory is allocated dynamically when an object or array is created, and it is deallocated when the object or array is garbage collected.

1. How can we handle errors in js?

Ans: There are a few ways to handle errors in JavaScript. One common method is to use the try... catch statement. This statement allows you to execute a block of code and catch any errors that occur during its execution. You can also include an optional finally block, which will always execute, regardless of whether an error was thrown or not.

1. What is event bubbling and capturing?

Ans: Event bubbling and capturing are two ways that events are handled in JavaScript.

Event bubbling is the default behavior of events in JavaScript. When an event occurs on an element, it is first processed by that element. Then, the event bubbles up to the parent element, and then to the parent's parent, and so on, until it reaches the document object.

For example, if you have a button inside a div, and you click on the button, the click event will first be processed by the button. Then, it will bubble up to the div, and then to the document object.

Event capturing is the opposite of event bubbling. When an event occurs on an element, it is first processed by the document object. Then, it captures down to the parent element, and then to the parent's parent, and so on, until it reaches the element where the event occurred.

For example, if you have a button inside a div, and you click on the button, the click event will first be processed by the document object. Then, it will capture down to the div, and then to the button.

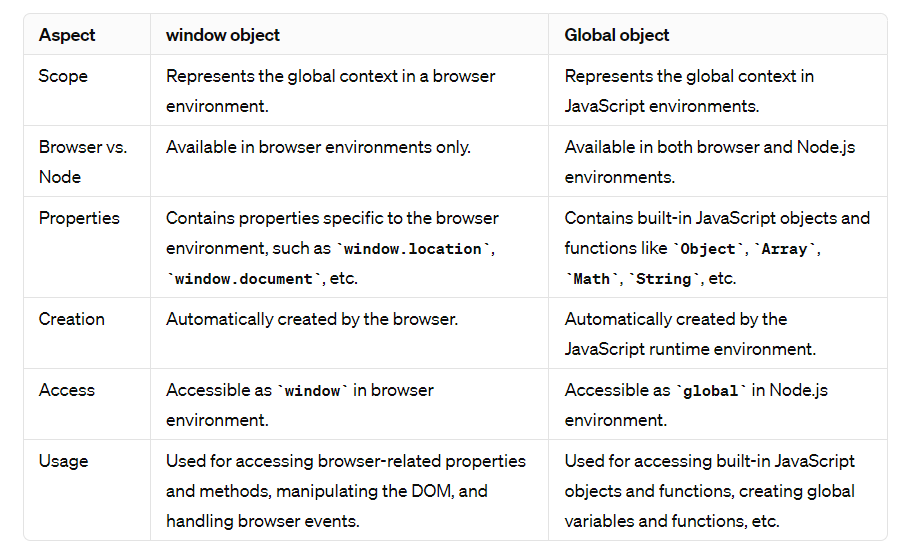
You can also use the stopPropagation() method to stop an event from bubbling up to the parent elements.

Event bubbling and capturing can be useful for handling events in complex web applications. For example, you can use event bubbling to handle events on child elements without having to add event listeners to each child element. You can also use event capturing to handle events on parent elements before they are handled by child elements.

1. What is an event?

Ans: An event in JavaScript is an action that occurs, such as a user clicking on a button, pressing a key, or moving the mouse over an element. When an event occurs, the browser generates an event object that contains information about the event. This event object is then passed to the event handler function, which is a block of code that is executed in response to the event.

1. Difference between window and global objects.

Ans: 

1. What is async and await?

Ans: async: The async keyword is used to define an asynchronous function. An asynchronous function returns a promise implicitly, allowing you to use await within it.

await: The await keyword can only be used inside an async function. It waits for a promise to resolve, and it pauses the execution of the async function until the promise is resolved or rejected. It then returns the resolved value.

1. What is fetch() and its uses?

Ans: The global fetch() method starts the process of fetching a resource from the network, returning a promise that is fulfilled once the response is available. The promise resolves to the Response object representing the response to your request.

1. What is module and its types?

Ans: A module in JavaScript is a file containing code that can be imported into other code files. JavaScript modules are an integral aspect of modern web development. These modules serve as containers for encapsulated pieces of code, allowing developers to organize their projects efficiently.

CommonJS modules:

These are the original way to package JavaScript code for Node.js. CommonJS modules are defined using the exports and require keywords.

ECMAScript modules:

These are the newer standard for JavaScript modules. ECMAScript modules are defined using the import and export keywords.

1. What is sort()?

Ans: The sort() function allows you to sort an array object by either the default sorting order, or by a custom sorting function.

By default, it sorts the elements in the array in ascending order based on their string Unicode values. The function takes the inputs, converts them to strings, and then sorts them using Unicode values.

1. What is class and constructor?

Ans: A class can be thought of as a blueprint that defines the framework for other objects to inherit. It defines the common methods and variables of all objects of a certain kind. Classes can be inherited, while constructor functions cannot.

A constructor is a special initialization function that is automatically called when a class is declared. It is a member function of a class, with the same name as the class name. The constructor automatically initializes the data members for an object of a class when an object of the same class is created.

1. What is the different types of OOP pillars we have?

Ans: Encapsulation is the process of wrapping data and code together into a single unit, called an object. This bundling of data and code together helps to protect the data from being accessed or modified by unauthorized code.

Abstraction is the process of hiding the implementation details of an object from the user. This allows the user to focus on using the object without having to worry about how it works.

Inheritance is the process of creating a new object that is based on an existing object. This allows the new object to inherit all of the properties and methods of the existing object.

Polymorphism is the ability of an object to take on many different forms. This allows the object to be used in a variety of different ways.

1. What keyword is used to inherit classes?

Ans: To create a class inheritance, use the extends keyword.

1. What is parent class and child class?

Ans: the class that derives from other classes is known as a derived or sub or child class. Whereas, the class from which the characteristic is derived is known as the base or super, or parent class.

1. What is object?

Ans: An object is a collection of properties, and a property is an association between a name (or key) and a value. A property's value can be a function, in which case the property is known as a method.

1. What is JSON.parse() and JSON.stringify()?

Ans: JSON.parse() takes a JSON string as input and converts it into a JavaScript object. This is useful for parsing JSON data that is received from a server or loaded from a file.

JSON.stringify() takes a JavaScript object as input and converts it into a JSON string. This is useful for sending JSON data to a server or saving it to a file.

1. What is JSON?

Ans: JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

1. What is function currying?

Ans: Function currying in JavaScript is a technique of transforming a function that takes multiple arguments into a sequence of functions that each take a single argument. This is done by partially applying the function, meaning that some of the arguments are fixed and the remaining arguments are passed to the new function.

1. What is set and map?

Ans: A Set is a collection of unique values. Each value can only occur once in a Set. Sets are similar to arrays, but they are unordered and cannot contain duplicate values. Sets can be used to store any type of data, including numbers, strings, objects, and even other sets.

A Map is a collection of key-value pairs. Each key can only occur once in a Map, but each value can occur multiple times. Maps are similar to objects, but they are ordered and can contain any type of data as keys or values. Maps can be used to store any type of data, including numbers, strings, objects, and even other maps.

1. Difference between set and array?

Ans: A screenshot of a white list

Description automatically generated

1. Difference between map and object?

Ans: A white background with black text

Description automatically generated

1. What is break and continue?

Ans: The break and continue statements in JavaScript are used to control the flow of loops. The break statement terminates the loop immediately, while the continue statement skips the current iteration of the loop.