1.What is hoisting?

It move the declaration of functions, variables or classes to the top of their scope, prior to execution of the code.

Hoisting allows functions to be safely used in code before they are declared.

2.. What is scoping?

Scope determines the accessibility (visibility) of variables.

JavaScript has 3 types of scope:

* Block scope –
* Function scope -
* Global scope

4.How are var, let const different?

|  |  |  |
| --- | --- | --- |
| **var** | **let** | **const** |
| The scope of a *var*variable is functional scope. | The scope of a*let* variable is block scope. | The scope of a *const* variable is block scope. |
| It can be updated and re-declared into the scope. | It can be updated but cannot be re-declared into the scope. | It cannot be updated or re-declared into the scope. |
| It can be declared without initialization. | It can be declared without initialization. | It cannot be declared without initialization. |
| It can be accessed without initialization as its default value is “undefined”. | It cannot be accessed without initialization, as it returns an error. | It cannot be accessed without initialization, as it cannot be declared without initialization. |

5. What are the two main differences in arrow functions?

Syntax change and in arrow function we cant use this.

6. Does Call apply bind work for arrow functions?

No,Because  "this" refers to a parent context and in arrow function there is no parent

* 7. What does call apply bind do?

The call() method invokes a function with a specified context. In other words, you can tie a function into an object as if it belonged to the object.

var obj = { num: 2 };function add(a){  
 return this.num + a;  
}

add.call(obj, 3);

The apply() method does the exact same as call(). The difference is that call() accepts an argument list, but apply() accepts an **array** of arguments.

|  |  |
| --- | --- |
|  | var obj = { num: 2 }; |
|  |  |
|  | function add(a, b){ |
|  | return this.num + a + b; |
|  | } |
|  |  |
|  | console.log(add.apply(obj, [3, 5])); |

Bind- instead of executing a function immediately, bind() returns a functionthat can be executed later on.

|  |  |
| --- | --- |
|  | Var obj ={num:2} |
|  | function add(a, b){ |
|  | return this.num + a + b; |
|  | } |
|  |  |
|  | const func = add.bind(obj, 3, 5); |
|  | func(); // Returns 10  7.What are closures?  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.  let name = 'John';  function greeting() {  let message = 'Hi';  console.log(message + ' '+ name);  } |

8. Debouncing?

=>A debounce is a cousin of the throttle, and they both improve the performance of web applications. However, they are used in different cases. A debounce is utilized when you only care about the final state. For example, waiting until a user stops typing to fetch typeahead search results

let name = document.getElementById("query").value;

if (name.length <= 2) {

return false;

}

console.log("fired");

let m = await searchMovies(name);

<input oninput="throttleFunction(main,1000)" type="text" id="query" />

Or

 const handleSearch=(e)=>{

        console.log(e)

        const filteredContacts = e.length === 0 ? db :

        db.filter(contact => contact.name.

                    toLowerCase().includes(e.toLowerCase()))

                    setfilter(filteredContacts)

    }

    const getData =()=>{

        axios.get("https://swiggy-list.herokuapp.com/rest").then(({data})=>{

            console.log(data,"kushagra")

            setdb(data)

            setloading(true);

        }).catch((err) => {

            setloading(false);

            setError(true);

          });

    }

9. Write a program to throttle a search bar?

Event bubbling?

**Event bubbling** is a method of event propagation in the HTML DOM API when an event is in an element inside another element, and both elements have registered a handle to that event. It is a process that starts with the element that triggered the event and then bubbles up to the containing elements in the hierarchy. In event bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

* **What is event loop?**

**=>In computer science, the event loop is a programming construct or design pattern that waits for and dispatches events or messages in a program. The event loop works by making a request to some internal or external "event provider", then calls the relevant event handler.**

**The event loop is the secret behind JavaScript’s asynchronous programming. JS executes all operations on a single thread, but using a few smart data structures, it gives us the illusion of multi-threading.**

**12.Explain promises to a 5 year old, with simple examples**

**=>a statement by a person that he or she will do or not do something I made a promise to pay within a month**

**13.what does async await mean?**

**=>Async/Await is the extension of promises which we get as a support in the language. You can refer Promises in Javascript to know more about it. Async: It simply allows us to write promises based code as if it was synchronous and it checks that we are not breaking the execution thread**

**14.What does the this keyword mean? =>**

**Within an instance method or a constructor, this is a reference to the current object — the object whose method or constructor is being called. You can refer to any member of the current object from within an instance method or a constructor by using this .**

**15.What are classes? what are getters and setters?**

**=>Getters and setters are used to protect your data, particularly when creating classes. For each instance variable, a getter method returns its value while a setter method sets or updates its value. Given this, getters and setters are also known as accessors and mutators, respectively**

## How do you declare private and static variables in classes

## =>static:As static variables are initialized only once and are shared by all objects of a class, the static variables are never initialized by a constructor. Instead, the static variable should be explicitly initialized outside the class only once using the scope resolution operator (::)

## private:Variables that are declared private can be accessed outside the class, if public getter methods are present in the class

## 17.What is currying?

## =>currying is the technique of converting a function that takes multiple arguments into a sequence of functions that each takes a single argument.

## 18.Write a program to flatten an array

## =>let flatArray = [].concat.apply([], arr); //Output: [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 ]