**--TOP JS QUESTION (Shreyansh Coding School)**

Js

Word vs keyword

Var const let

The difference

Window object

Browser contecxt Api

Stack

Heap memory

Execution context

Lexical environment

Hoisting

Types in js

How to copy refernce values

Conditiona;

If else else if

Truthy vs falsy

Switch

1. **Words vs keyword ?**

Compiler – compiler is a guy jo code ko chalata hai.

Words – word is that thing that doesn’t have any meaning in JS.

Ex- bhai, hello, suresh, raj12

Keywords: keyword is reserve word in js that have meaning in js and use for specific purpose

Ex- for (use in loop) , var (variable) , switch (switch case) , if , else , let etc .

**2)Whar is var , const , let in Js ?**

Already read it in Js interview series

Let is a block or braces(cruely braces k under hi value accessible hai) scope.

Const value can not be changed after declare –

Var is global and functional scope

Var introduced in es5 where let and const introduced in ES6.

Var add itself to the window object.

Where let const doesn’t .

Proof : 1) write in vs code – var a=12; Let b =15; )

2)Now go to console and type – window (enter press)

3)Now window pe click u will see a=12 ; add hai but let wala b=15 add nahi hai .)

**Ex - Const**

const name = “rahul”;

Console.log(name) //output : rahul

Const name = “abhi”;

Console.log(name)

//output : error comes (coz name variable is cannot modified in const case)

1. **What is Hoisting ?**

Already read in js interview notes

Var a = 10;

Var a; //declaration of vode

a=10; //initialization

note : in hoisting declarartion moved to top .

1. What are data type in JS?

already read in Js interview notes (primitive and Non primitive data type)

The reference data type is – the value comes under - [] (Array) , {} (object) , () function.

Reference value – aisi value jiska copy karne real value copy nhi hota hai uski reference copy ho jata hai.

1. **What is object ?**

In JavaScript, an object is a fundamental data structure that represents a collection of key-value pairs,

Its is used to stor collection of data.

When U have to store all details of Single person or thing then we used a object.

Ex - const person = {

firstName: "John",

lastName: "Doe",

age: 30,

email: "john@example.com",

};

1. **What is conditional operator ?**

If else elif conditional statement already read in js Classmate copy .

1. What is loop in Js ?

For , while , do while (already read in Js classmate notes)

1. Version of JS – ES5 (Var) , ES6 (let and const introduced in Es6)

**6)What IS function in JS ?**

Function is use to reuse the code

When u have run same logic type of code but with different value everytime.

When u have to run code not immediately , but not in future.

Ex –

Function Sum(a,b,) {

Var a= a;

Var b = b;

Var c = a+b’

}

Sum(5,6) //Output : 11

Sum(7,8) //Outpu: 15

Sum(2+3) //output : 5

**9)Whar is Array and its method ?**

Already mention in JS Classmate notes

Read- push ,pop, shift ,unshift, splice method.

**ADVANCE JAVASCRIPT COURSE**

10)

function abc() {

for(var i=0; i<5; i++) {

// console.log(i); //output: 0,1,2,3,4

}

console.log(i); //output: 0,1,2,3,4 or 5

}

abc()

**11) what is window object ?**

window object is that where browser related feature is store. whc we can use in JS . window is a box of feature wch is given by browser .

Js use some feature that is not part of js but still we can use in js by window object , open inspect and go to console –

Type : window (press enter)

These are not part of Javascript . its part of window object .( TO check : go to browser ,

ex – alert()

prompt()

console() , print()

setTimeout() , setinterval()

localstorage , find() , fetch()

IN Browser pre in built html, css ,js install hote hai.

**12)What is browser context API ?**

Browser give some feature wch is in window object

Browser provide three thing –

1. Stack ,
2. heap memory (all data which we create in js store in heap memory , or intermediate data

ex- if we want to add – 1+2+3+4+5

computer add this one by one example – computer calculate first 1+2 =3 now 3 is intermediate data i.e its not final answer so 3 store in heap memory , now computer take 3 form heap memory and add with 3 – > 3+3 =6 (now 6 store in heap) then computer similarly take 6 form heap memory and calculate 6+4 = 10 , (now 10 store in heap) then computer final calculate 10+5 = 15 final output .}

this way computer work !

1. window object .

Note : The combination of 3 thing is Context api .

**13) what is execution context and lexical enviorment?**

When we run or called a any function - abc() then function create its own imaginary container , this container is called execution context, where the function whole code written is executed .

where three thing are stored in this container–

1. all variable value
2. function inside that parent function (function k under if we create another function that function is store)
3. lexical environment(it tell function wch variable is acces or wch variable is not) of that function.

**Ex3**-

1 function abc() {

2 var x= “hello”

3 function def() {

4 var name = “suresh”

5 console.log(name);

}

}

abc()

**Note**: here if we want to use name variable in parent function abc k loc3 se pahle then it will throw error coz var is functional scope and its value is accessible inside his parent function wch is def . dts y its not accessible . so is chiz ko batane ka kaam basically Lexical environment ka hai.

**It’s a chart where written your particular function is allow to access which thing or which not** .

In fancy word ,its contain scope or scope chain .

**14) How to copy reference value?**

Method1 ->

var a = [1,2,3,6,7]

var b = a; //copy a value

console.log(b)

b.pop() //remove last element from b

console.log(b)

console.log(a);

Note : this make problem check this vs code filE , So its wrong method.

file - reference\_Value.js

Method2: by use of spread operator we copy reference value-

//By Spread OPerator WE copy reference value of array!

var x = [1,2,3,4,5]

var y = [...x];

console.log(y) //output [1,2,3,4,5]

y.pop()

console.log(y);         //output : [1,2,3,4]

console.log(x);          //output: [1,2,3,4,5]

**15) How to copy object value ?**

Var person = {

name: “harsh”,

age:26

}

Var copyObj = {…person}

Console.log(copyObj)

copyObj.name = “Anshika”;

console.log(copyObj);

console.log(person);

**16) what is truth and falsy ?**

Falsy value : 0 ,false, undefined , null , Nan, document.all

Truthy : rest all are truth value (-1 also is truthy value coz its not belong to falsy value) and in this case if condition executed.

Ex- (8>7) , (7=7) , (6<5)

If(true) {

Console.log(“hello”)

}

**NOTE: //if if k under true value then if condition executed**

**EX2-**

If(false) {

Console.log(“hello”)

} else() {

Console.log(“world”)

}

**Note : here else condition value executed .**

**what is Switch , continue , break case ?**

its already we read in Js classmate notes

**9)What is forEach loop ?**

In JavaScript, the forEach loop is a method available for arrays. It is used for iterating over the elements of an array and executing a provided callback function once for each element. The forEach loop is a convenient and concise way to perform operations on each element of an array.

Ex – var array1 = [1,2,34,56,21,6,7.5];

Array1.forEach((num) => {

Console.log(num+2);

}

);

Output: [3 ,4,36,58,23,8,8.5 ]

Note: add 2 in each elemet of array value me.

**10)what is forIN loop in js?**

The for In loop for used Object in Javascript.

The for...in loop in JavaScript is used to iterate over the enumerable properties of an object. It allows you to loop through the keys(property names) of an object and perform a specific action for each key.

or access the corresponding values associated with those keys .

Ex –

const person = {

firstName: "John",

lastName: "Doe",

age: 30, };

for (let key in person) {

console.log(key + ": " + person[key]);

}

**Output :**

firstName: John

lastName: Doe

age: 30

Note: if only write -

console.log(key)

output :

first name

last name

age

**12) how array are made behind the scene ?**

in Java, an array is considered an object. It is a dynamically created object that can hold elements of the same data type.

Proof : type In console – typeof []

output : object .

**how exactly made array behind the scene** –

var array1 = [1,2,3,4,56]

in internal array is converted into object like this (where 0 index pe 1 , 1 index pe 2 , 2 index pe 3 and so on……..) – >

arr = {

0 : 1,

1: 2,

2 : 3,

3 : 4,

4 : 56

}

**Note : To check when it is array or when it is object in JS** –

Array.isArray([]) - true (output: true i.e. it is array)

Array.isArray({} ) - false (output: false give i.e it is object)

**13)How to delete object props?**

var a = {

Name: “harsh”,

Age: 24,

}

delete a.age;

delete a.name;

console.log(a);

Output : {} (empty object comes)

**MASTER JAVASCRIPT ADVANCE COURSE**

**JAVASCRIPT IS A SYNCHRONOUS, it’s a Single threaded language.**

**One computation at a time, ek baar me ek hi function pe kaam ho raha hota hai.**

**Synchronous**: when we have 3 or multiple code or programme its executed one by one (after finishing then next start,tab tak he wait for his chance) is called synchronous –

Ex – 1) Bootle me paani bharna

2) market se maggi lana

3) 1 movie download

Synchronous is when u do this all work one by one –

First, bottle me paani bhara

Second paani fill hone k baad mai market se maggi laya

Third me Maggi lane k baad 1 movie download kiya.

**Asynchronous**: Asynchronous allow your code to run in the background without blocking the execution of other time . (ex-setTimeout).

Let suppose that if we get data from the server through Api then isko Async banana padega dts y we use - fetch(“htttp://fb.com”)

Taki ye iske baad wale code ko block nhi kare execute hone se. Coz in Js code run line by line.

when all code or program start at the same time and run simultaneously, it doesn’t wait to finish the previous code then then start second. Coz some code ka pata nahi hota wo kab run hoga , ex – fetch data form server through API .

Ex- bottle me paani bharne k liye laga diye , uske baad 1 movie download me laga diye , then market gaya maggi k liye .

Now market se maggi le k aaya so ist task done ✅

Aa k dekha paani bhar chuka tha so second task also done ✅

Now after 1 minute movie also been downloaded so third task also done ✅

Note: Asynchronous takes less time as compared to synchronous. code doesn’t wait here, its runs simultaneuosuly . its doesn’t block execution of another code.

Ex of Asynchronous function –

setTimeout , setInterval, promises , fetch, axios , XMLHttpRequest.

Note :if function is not any type of this givenn then we can say that its a synchronous function.

**11)what is Callback function in JS?**

A callback function in JavaScript is a function that is passed as an argument to another function and is executed after the parent or outer function has completed its task.

Ex- callback\_function.js

Callback function is used for that function wch make take some time to complete and after completion its tell to main function its now ready to run.

Asynchronous function (it is callback function) –

setTimeout ( function() {

console.log(“Hi I am sid , mai 2 sec baad print hua hu”);

} , 2000);

Output : (comes after 2 sec of count is ) -

Hi I am sid , mai 2 sec baad print hua hu

Note: aisa code jo baad me chalta hai use ek function bana k setTimeot k under daal dete hai. i.e. called callback function.

**15) What is CallStack in JS ?**

Callstack is a List where function is stored and it track a record of all function which execute first and which execute after that function and which execute in last .

Its play a crucial role In managing the execution of code and function call .

**16) what is Even loop in Js ?**

There are two thing – main stack (callstack) and Side stack (Event queue) .

Jo v Main stack(Call stack) par hai if it’s a synchronous it will give output. Synchronous code always go to Main stack. Where Asynchronous go to the side stack.

Ex- Event\_Loop.js

Jo v function Side stack (Event Queue) pe wo behind the scene process karta after time complete it go to main stack that is call stack and its also give output.

**Even loop main stack me side stack me chize lata , after side stack function time is over and when main stack is empty to run this code .**

**17) What to use to write Async code ?**

Fetch , axios , promises , setTimeoit , setInterval , XMLhttprequest.

After get data from server or somepoint of time –

Answer chalane k liye -

then catch

async await , callback

**18) What is promises in Js ?**

*The solution of callback hell is promise .*

A Promise is a JavaScript object that represents the eventual completion (or failure) of an asynchronous operation and its resulting value.

Promises are commonly used for handling asynchronous code or operations, wch make some time to complete, such as making network requests or fetch Api, reading files, or interacting with databases, in a more organized and manageable way.

It is used to find out the asynchronous operation is successfully completed or not. It is parallely executed in the background.

Promise is a type of constructor function that’s why we use new keyword before Promise word .

A promises is a object. There are three state.

1)pending. (intial state or process state)

2) resolved: The state represent asynchronous operation has completed successfully, and the Promise has a resolved value. represented by resolve().

( if resolved then block of code run)

3)rejected. The state represent asynchronous operation has encountered an error or failed, (if reject catch block of code run where reason of failure.) represented by reject().

Ex – Promises.js , Promises2.js

**When u write async code then it run in future the possibliy of this code is run or will not run (error comes )**

**19) What is Async Await in Js ?**

async/await is a feature in JavaScript that makes it easier to work with =asynchronous code, particularly when dealing with Promises. It allows you to write asynchronous code in a more sequential and readable manner, resembling synchronous code, which can make your code more maintainable and less prone to callback hell, also known as the "Pyramid of Doom.

In Simple words , we can say that async-await is used taki response jo aaya return me us answer ko dikhane k liye .then ka use naa karna pade.

Without Async await –

Function abcd() {

fetch(`https://randomuser.me/api/`)

.then(function (data) {

return data.json(); //data convert into json format

})

.then(function(data) {

Console.log(data);

});

}

abcd()

With Async Await – (it's so easy)

1 Async function abcd() {

2 let data = await fetch(`https://randomuser.me/api/`)

3 let answer = await data.json();

4 Console.log(answer);

}

}

abcd();

//loc2 after comes raw data, Loc3 convert into json format,

await use cozl oc4 wala code tab tak nahi chalege ya print karega jab tak data convert naa ho!

**20) What is Concurrency and parallelism ?**

In Js async and sync code both are processes and run together that is called concurrency (main stack and side stack work at the same time).

Concurrency refers to the ability of a system to handle multiple tasks or processes seemingly simultaneously. It doesn't necessarily mean that these tasks are executing at the exact same time, This allows the code to continue executing while waiting for other operations to complete. but rather that they are making progress independently and can potentially overlap in their execution.

JavaScript achieves concurrency through features like callbacks, Promises, and async/await, which allow for the execution of tasks in parallel or in a non-blocking manner.

Where as parallelism related to processor core (in one processor there are multi core – dual core , octa core ,quad core and each core process one calculation or one thread hold, parallelism means you can do a work with your different – different cores )

**21) What is the difference between local storage and sessional storage?**

Local Storage and Session Storage are both web storage options that allow web applications to store data locally on a user's device.

It available in modern web browsers However, they have some key differences:

1. \*\*Data Persistence:\*\*

- \*\*Local Storage:\*\* Data stored in local storage persists(rahna) even after the browser is closed and is not limited by a session. It remains available until it is removed or cleared by the user or the web application.

- \*\*Session Storage:\*\* Data stored in session storage is only available for the duration of the page session or when user logged ic n page. It gets cleared when the user closes the browser tab or logged out i.e. website(web page) or exits from the page. .

2. \*\*Storage Capacity:\*\*

- \*\*Local Storage:\*\* Typically, local storage offers a larger storage capacity compared to session storage. It can store more data, making it suitable for long-term storage needs.

- \*\*Session Storage:\*\* Session storage has a smaller storage capacity compared to local storage. It's designed for storing temporary data that is only needed for the duration of a user's visit.

3. \*\*Scope:\*\*

- \*\*Local Storage:\*\* Data in local storage is accessible across multiple tabs or windows of the same browser from the same origin (domain). It's not limited to a single tab or window.

- \*\*Session Storage:\*\* Data in session storage is scoped to the specific tab or window from which it was set. It's not accessible to other tabs or windows, even if they are from the same origin.