Intake Questions

Questions Asked by: Emil J S , Muhammed Jaz , Vipin Varghese , Hariprasad , Jaseel (QA Team) , Vishnu GR

Java script

Theory :

- event propagation,
- event delegation,
- pass by value vs pass by reference,
- regular function vs arrow function
- arrow function vs regular function (this)
- is it possible to use arrow function as a constructor?,
- object freeze vs seal,
- Object literals
- Shallow copy
- Deep copy
- Deep copy vs Shallow copy
- applications & drawbacks of closure
- benefits of IIFE
- map vs forEach
- nullish operator
- object methods
- hoisting in let and const
- Type error vs referance error
- error handling in javascript
- factory function
- console.log({} == {}); find the answer
- non primitive datatype

- promise.any
- why callbacks are used?
- does null have prototype?
- when does ReferenceError happen?
- Use of Object.entries

Practical:

- toggle checkbox by button click
- prevent right click of a button
- find common elements from 2 arrays
- remove last property from object
- find sum of keys of an object
- find total price of products
- Object.keys
- Object.values
- highest even number from an array using reduce
- how to create a class
- object manipulation
- find sum of values in an object
- callback
- sum of elements in given array
- Array.reduce: sum of positive numbers in a mixed array
- check if object is empty
- throwing errors
- countdown timer: 10 to 0
- generate random number between 0 and 10
- current time, without date
- Slice
- currying
- call

- apply
- bind

Node js

Theory :

- node concurrency,
- child process,
- how to check the existence of a middleware,
- application middleware vs router middleware,
- app.use vs app.set
- app.all
- app.set
- http request structure
- Parts of HTTP request and response
- HTTP headers
- dynamic routing,
- router chaining,
- CORS
- CSRF
- content negotiation,
- web api vs rest api
- REST principles,
- fork vs spawn
- error first callback
- httponly
- body parser
- post vs put
- http OPTION
- same origin policy

- query params vs path params
- CommonJS
- reading query params and path params
- path params
- catch-all route
- user-agent
- HTTP 400
- sending status code

Practical:

- middleware to log all parameter names
- how to setup a router level middleware

MongoDB

Theory :

- covered query
- drawbacks of indexing
- $pop vs $pull
- journaling
- Exec vs Exec file
- Clustered collection
- Clustered index
- mongoose
- ODM
- Accumulators examples
- Mongo DB Utilities
- Denormalization
- Trigger in mongoDB

- Durability (ACID)

Practical:

- query - avg mark in class 10
- query - second largest mark in class 10
- query - increase all marks by 10%
- $group
- $max
- $min
- $expr
- $elemMatch
- $exist
- facet
- bulkwrite
- lookup
- add field to an object in mongodb
- createView
- reduce salary by 10%
- names ending with "j"

1. Array reverse - for loop while loop without using extra array...
2. Unique
3. Prime
4. Duplicate
5. Sort - ascending, descending
6. Post pre increment
7. Logical function
8. Operators
9. Keywords

10. Identifiers

11. Variable

12. Datatype

1 find average of even numbers

2 create a function and 2 parameters arr and str nnt str length even num anel arrayile evensinte sum and thirichum

Sum of 2 numbers = target

Find Greater in 3 variables

Sum using callback

Leetcode

Generator function - multiplication table

Recurssion

Function to check prime number or not

do while loop

let a=10

let b=a++

let c=++a

console.log(a,b,c)

let a = 10;

let b = ++a; // Pre-increment: a becomes 11 first, then b gets 11.

let c = a;

sort array using while loop

withot another array

find a frequncy

find unique number

```
let arr=[1,2,3,4,5,6,2,32]

let freq={}

for(let i of arr){
   freq[i]=(freq[i] ||0)+1
}

console.log(freq)

for(let [key ,value ] of Object.entries(freq)){
   if(value==1){
      console.log(key)
   }
}
```

unsorted array find prime number and sort it

*second largest

*break and continue

*negavite and positive even and odd and delete odd and even number replased 00

Theory

Data types

Variables endhina

Functions endina

1.Create array

2.Add into another array without push

3.Add fizz if multiple of 3 and buzz if multiple 5 and fizzbizz for both multiple

4.Insert in same array without inbuilt method

5.A++

++A

6.Reverse array without inbuilt(swap)

7.add two element from the array find pair which equals the target

8.multiply all numbers with 2 and add into another array

9.multiply add the number which is multiple of 2 and add into array

10.print 20 to 2 even number only using while loop (with if and without if)

```
 function a(aa,bb,c){
    console.log(c(aa,bb))
}

function b(aa,bb){
    return aa + bb
}

let aa = 10
let bb = 5
a(aa,bb,b)
let ar = [1,2,3,4,5]



for(i = 0;i < ar.length ; i++){
    for(j = i+1; j < ar.length ; j++){
        if(ar[i] + ar[j] === 8){
            console.log(ar[i],ar[j])
            break
        }
    }
}
let ar = [1,2,3,4,5]
let res = []
```

```
let j = 0


for(i = 0;i < ar.length ; i++){
   if(ar[i] % 2 === 0){
      res[j] = ar[i] * 2
      j++
   }
}
console.log(res)
let i = 20
while(i >= 2){
  console.log(i)
  i -= 2
}
```

2 array merging without using inbuilt method .

2array ...1st array il odd and 2 nd array even store cheyya without using newarray

Do while using multiple of 5

\

```
// let num =100;
// let i=1

// while (num>i){
//     console.log(num)
//     num-=2
// }



// find squir of the numbers
// let arr=[1,2,3,4,5,6,7]

// let sqr=[]
```

```
// sqr=arr.map(num => num*num)
// console.log(sqr)




// find frist even number
// let num =[1,5,3,8,4,9]

// for(let i=0; i<num.length; i++){
//     if(num[i] % 2 ==0){
//         console.log('this is the firs even number',num[i])
//         break;
//     }
// }



// leecode question (412)

// var fizzBuzz = function(n) {
//     let output=[]
//     for(let i=1; i<=n; i++){
//         if(i % 3 ==0 && i % 5 ==0){

//             output.push('FizzBuzz')
//         }else if(i % 3 == 0){

//             output.push('Fizz')
//         }else if(i % 5 == 0){
```

```
//        output.push('Buzz')
//    }else{
//        output.push(i.toString())
//    }
//    }
// return output
// };
```

Practical

1 .Delete odd number without using inbuild methods and new array

2 .break and continue (guess the output)

3 .find unique

5 .delete duplicate (without inbuild and new array)

6 .increament and decreament (guess the output)

Theory

Oops concept

```
// let arr=[1,4,6,3,2,5]
// let even=[]
// for(let i=0;i<arr.length;i++){
//    for(let j=0;j<arr.length;j++){
//        if(arr[j]>arr[j+1]){
//            let temp=arr[j]
//            arr[j]=arr[j+1]
//            arr[j+1]=temp
//        }
```

```
//     }
// }
// let index=0

// for(let i=0;i<arr.length;i++){
//     let odd=false
//     for(let j=0;j<arr.length;j++){
//     if(arr[i]%2!=0){
//       odd=true
//      break;
//     }
//     }
//     if(!odd){
//         arr[index]=arr[i]
//         index++
//     }



// }
// arr.length=index
// console.log(arr)



// let i=1
// do{
//     console.log(i*5)
//     i++
// }while(i<=10)
```

```
for (let i = 1; i <= 10; i++) {
   console.log(++i);
   continue

   if (i == 5) {
     break;
   }
   console.log(i--);
   continue;
 }
```

Sort array

Delete odd without new array and build in method

Do while

Guess the output

```
let arr=[];
for(let i=0;i<=20;i++){
   arr[arr.length]=i
}
console.log(arr)

let even=[];
let odd=[];
let eindex=0;
let oindex=0;
for(let i=0;i<arr.length;i++){
if(arr[i]%2===0){
   even[eindex++]=arr[i]
}else{
   odd[oindex++]=arr[i]
}}
```

```
console.log(even)
console.log(odd)
 let obj = {
    d: 3,
    a: 1,
    c: 4,
    b: 2
};
let largest = -Infinity;
let secondLar = -Infinity;
let largeKey;
let secondLarKey;

for(let key in obj) {
    if(obj[key] >= largest) {
        secondLar = largest;
        largest = obj[key];
        secondLarKey = largeKey;
        largeKey = key;
    } else if(obj[key] >= secondLar) {
        secondLar = obj[key];
        secondLarKey = key;
    }
}
delete obj[secondLarKey];
console.log(obj)
```

Here are 50 algorithm questions that are simple enough to strengthen your foundation:

1. Prime Number Check

Write a program to check whether a number is prime or not.

2. Even Numbers from 100 to 0

Print even numbers from 100 to 0 using a while loop.

3. Sum of Numbers Using While Loop

Find the sum of numbers from 1 to N using a while loop.

4. Reverse an Array

Reverse an array using a for loop.

5. Palindrome Check

Check whether a given string or number is a palindrome.

6. Factorial of a Number

Write a program to find the factorial of a number.

7. Fibonacci Series

Print the Fibonacci series up to the Nth number.

8. Count Digits in a Number

Find the number of digits in a given number.

9. Reverse a String

Reverse a string without using built-in methods.

10. Find the Largest Number in an Array

Find the largest number in an array.

11. Find the Smallest Number in an Array

Find the smallest number in an array.

12. Count Vowels in a String

Count how many vowels are there in a string.

13. Count Even and Odd Numbers in an Array

Count how many even and odd numbers are there in an array.

14. Find Sum of Digits of a Number

Find the sum of digits of a given number.

15. GCD (Greatest Common Divisor)

Find the GCD of two numbers.

16. LCM (Least Common Multiple)

Find the LCM of two numbers.

17. Check for Armstrong Number

Check if a number is an Armstrong number.

18. Merge Two Arrays

Merge two arrays into one.

19. Check for Leap Year

Check whether a given year is a leap year.

20. Find the Second Largest Element in an Array

Find the second largest element in an array.

21. Remove Duplicates from an Array

Remove duplicate elements from an array.

22. Find the Missing Number in an Array

Find the missing number from an array containing numbers 1 to N.

23. Reverse Words in a Sentence

Reverse the words in a sentence, not the individual characters.

24. Sum of Even Numbers from 1 to N

Find the sum of even numbers between 1 and N.

25. Find the Frequency of Elements in an Array

Count the frequency of each element in an array.

26. Find the Middle Element of an Array

Find the middle element of an array.

27. Check if a String is an Anagram of Another

Check if two strings are anagrams of each other.

28. Check if a Number is Perfect

Check if a given number is a perfect number.

29. Sum of All Positive Numbers in an Array

Find the sum of all positive numbers in an array.

30. Remove All Negative Numbers from an Array

Remove all negative numbers from an array.

31. Print a Pattern of Stars

Print a pattern like a triangle or diamond with stars (*).

32. Check if a Number is Even or Odd

Check whether a number is even or odd.

33. Find the Average of Numbers in an Array

Find the average of all numbers in an array.

34. Swap Two Numbers Without Using a Third Variable

Swap two numbers without using a temporary variable.

35. Count the Occurrence of a Character in a String

Count how many times a character appears in a string.

36. Find the Sum of All Odd Numbers from 1 to N

Find the sum of odd numbers between 1 and N.

37. Find the Longest Word in a Sentence

Find the longest word in a sentence.

38. Convert a String to Lowercase

Convert a string to lowercase without using built-in methods.

39. Check if a String is a Substring of Another

Check if a string is a substring of another string.

40. Remove Spaces from a String

Remove all spaces from a string.

41. Find the First Non-Repeated Character in a String

Find the first character that does not repeat in a string.

42. Find the Sum of the First N Natural Numbers

Find the sum of the first N natural numbers.

43. Check if a String Contains Only Digits

Check if a string contains only digits.

44. Sort an Array in Ascending Order

Sort an array in ascending order without using built-in sort methods.

45. Check if a Number is Divisible by 3 and 5

Check if a given number is divisible by both 3 and 5.

46. Find the Largest Palindrome in an Array of Strings

Find the largest palindrome from an array of strings.

47. Check if a Number is a Perfect Square

Check if a given number is a perfect square.

48. Find the Sum of Diagonal Elements in a 2D Array

Find the sum of diagonal elements in a 2D array (matrix).

49. Print the Multiplication Table of a Given Number

Print the multiplication table for a given number.

50. Find the Frequency of Each Character in a String

Count the frequency of each character in a string.

theory

Function

Data type

Variable

Identifies

practical

Push an array in to an empty array without using push method

Post increment and pre increment

Callback function

Reverse an array

Practical using && and ||

delete element from middle of an array

```
 function removeDuplicates(str) {
  let count = {};
  let result = "";
  for (let i = 0; i < str.length; i++) {
     let char = str[i];
     count[char] = (count[char] || 0) + 1;
  }
  for (let i = 0; i < str.length; i++) {
     if (count[str[i]] === 1) {
        result += str[i];
```

```
    }
  }
  return result;
}
let s = "hello world";
console.log(removeDuplicates(s));
 function reverse(arr){
  let word = "";
  let res = "";

  for(let i = 0 ; i<str.length ; i++){
    if(str[i] !== " "){
      word = str[i]+word;
    }else{
      res += word + " ";
      word ="";
    }
  }
  res += word;

  return res
}


let str = "Hello World";
console.log(reverse(str))
 function check(str) {

    let max = -Infinity;
    let maxkey = null;
    let count = {}
```

```
    for (let x of s) {
        count[x] = (count[x] || 0) + 1;
    }



    for (let z in count) {
        if (count[z] > max) {
            max = count[z];
            maxkey = z
        }
    }
    let allone = (Object.values(count).every(val => val === 1))


    return { maxkey, max, allone }
}
let s = "javascript";
let res = check(s);
let ar = [1, 2, 3, 4, 5, 4]
for (i = 0; i < ar.length; i++) {
  for (j = i + 1; j < ar.length; j++) {
    if (ar[i] + ar[j] === 8) {
      console.log(i, j)
      break
    }
  }
}
 function a (arr){
  for(let i = 0 ; i<arr.length ; i++){
```

```
    if(arr[i] < 2){

      for(let k = i ; k<arr.length-1 ;k++){
        arr[k] = arr[k+1];
      }
      i--;
      arr.length--;
      continue;
    }
    let p = true;
    for(let j = 2 ;j*j <= arr[i] ; j++){
      if(arr[i]%j === 0){
        p = false;
        break;
      }
    }

    if(!p){
      for(let k = i ; k<arr.length-1 ;k++){
        arr[k] = arr[k+1];
      }
      i--;
      arr.length--;
    }
  }

  return arr
}



let arr = [1,2,3,4,5,6,7,8,9,0];
```

```
console.log(a(arr));
 function unique(arr){
  let n = arr.length;
  let index = 0;
  if(n<=1) return arr;

  for(let i = 0 ; i<n ; i++){
    let count = 0 ;

    for(let j = 0 ; j<n ; j++){
      if(arr[j] === arr[i]) count++
    }

    if(count === 1) arr[index++] = arr[i];
  }

  arr.length=index;
  return arr;
}

let arr = [1,2,3,4,5,6,7,7,5,8,8];
console.log(unique(arr))
function removeDuplicates(arr){
  let n = arr.length;

  for(let i = 0 ; i<n ; i++){
    for(let j = i+1 ; j<n ; j++){
      if(arr[j] === arr[i]){
        for(let k = j ;k<n-1 ; k++){
          arr[k] = arr[k+1];
        }
```

```
        n--;
        j--;
      }
    }
  }

  arr.length = n ;
  return arr;
}

let arr = [1,2,2,1,3,4,4,5,5,6,7,8,9,0]
console.log(removeDuplicates(arr))
function sort(arr){
  let n = arr.length;
  for(let i =0 ; i<n ; i++){
    for(let j = 0 ; j<n ; j++){
      if(arr[i]<arr[j]){
        [arr[i],arr[j]]=[arr[j],arr[i]];
      }
    }
  }
  return arr;
}

function check(arr){
let arr2 = sort(arr),start = arr2[0],n=arr2.length,end=arr[n-1],i=0;

while(start<=end){
  if(arr[i] === start){
    i++
  }else{
```

```
    console.log(start)
    // arr2[i]=start;
    // i++
  }
  start++;
}
return arr2;
}

let arr = [1,2,3,5,7,9,11];
console.log(check(arr));
function reverseString(str) {
    let reversed = "";
    let i = str.length - 1;

    while (i >= 0) {
        reversed += str[i];
        i--;
    }

    return reversed;
}

let str = "hello world";
console.log(reverseString(str));
console.log(x++)
console.log(++x)
console.log(x--)
console.log(x+1)
console.log(--x)
```

OBJECT Manipulation QUESTIONS

1.find the largest and second largest from the object! 🚀

```
let obj = { a:10, b:20, c:30, d:40 };

let large = -Infinity;    // For maximum value
let seclarge = -Infinity; // For second maximum value

for (let key in obj) {
   if (obj[key] > large) {
      seclarge = large;       // Old max becomes second max
      large = obj[key];      // New max found
   } else if (obj[key] !== large && obj[key] > seclarge) {
      seclarge = obj[key];   // Update second max
   }
}

console.log(large);    // 40
console.log(seclarge); // 30
```

2.Find the sum of values from the object?

```
let obj={
   a:10,
   b:20,
   c:35,
   d:5,
}
```

```
let sum=0;

for(let key in obj){
   sum=sum+obj[key];
}
console.log(sum)
```

3.Find the largest value and the key?

```
let obj={
   a:40,
   b:20,
   c:35,
   d:5,
}
let largest = -Infinity
let largekey =""

for(let key in obj){
   if(obj[key]>largest){
      Largest = obj[key]
      Largestkey = key

   }
}
console.log(largest)
console.log(largestkey)
```

4.Print keys and values in the object?

```
let obj={
   a:40,
   b:20,
   c:35,
   d:5,
}

for(let key in obj){
   console.log(key,obj[key])

}
```

5.Find sum of even values?

```
let obj={
   a:2,
   b:5,
   c:10,
   d:8,
}
let sum=0;
for(let key in obj){

   if(obj[key]%2===0){
      sum=sum+obj[key];
```

```
    }

}

console.log(sum)
```

6.Find sum of odd values?

```
let obj={
    a:2,
    b:5,
    c:10,
    d:7,
}
let sum=0;
for(let key in obj){

    if(obj[key]%2!==0){
        sum=sum+obj[key];
    }

}
console.log(sum)
```

7.Find all keys whose values>10?

```
  let obj={
   a:9,
   b:45,
   c:34,
   d:90,
    }

  for(let key in obj){
     if(obj[key]>10){
   console.log(key)
      }
  }
```

```
let obj={
   a:9,
   b:45,
   c:34,
   d:90,
}
let swap={}
```

```
for(let key in obj){
   let value=obj[key];
   swap[value]=key
}
console.log(swap)
```

8.Count the number of keys in an object.

```
let obj={
   a:9,
   b:45,
   c:34,
   d:90,
}

let count=0;

for(let key in obj){
   count++;
}
console.log(count)
```

9.Create a new object containing only those entries where value > 20.

```
let obj={a:10,b:30,c:45,d:9,e:87};

let newobj={}
```

```
for(let key in obj){
   if(obj[key]>20){
      newobj[key]=obj[key]
   }
}
console.log(newobj)
```

10.Reverse the object (last key becomes first, second-last becomes second, etc.).

```
let obj={a:10,b:20,c:90,d:45};

let keys=[];
for(let key in obj){
   keys.push(key)
};

let reversedobj={};

for(let i=keys.length-1;i>=0;i--){
   let key=keys[i]
   reversedobj[key]=obj[key]
}
console.log(reversedobj)
```

OR

11.Check if a value exists in the object (without using object.Values())?

```
let obj={a:10,b:20,c:90,d:45};

let isfound=false
let searchvalue=45
for(let key in obj){
   if(obj[key]===searchvalue){
      isfound=true;
      break;
   }
}
console.log(isfound);
```

12.Find the average of all values in an object.

```
let obj={
   a:10,
   b:90,
   c:88,
   d:9,
}
let sum=0;
let count=0;
 for(let key in obj){
```

```
    count++
    sum=sum+obj[key];

}
console.log(sum)
console.log(count)
let avg=sum/count
console.log(avg)
```

13.Change all values to their double (example: 10 ➔ 20).

```
let obj={
   a:10,
   b:90,
   c:88,
   d:9,
}

for(let key in obj){
   obj[key]=obj[key]*2;
}
console.log(obj)
```

14.Delete all keys where value < 15?

```
let obj={a:10,b:34,c:90,d:8};
```

```
for(let key in obj){
   if(obj[key]>15){
      delete obj[key]
   }
}
console.log(obj)
```

15.Merge two objects into a single object?

```
let obj={a:4,b:3};
let obj2={c:34,d:6};

let merge={};

for(let key in obj){
   merge[key]=obj[key];
}

for(let key in obj2){
   merge[key]=obj2[key];
}
console.log(merge)
```

Or

```
let obj1={a:4,b:3};
let obj2={c:34,d:6};
```

```
for(let key in obj2){
   obj1[key]=obj2[key]
}
console.log(obj1);
```

16.Check if a key exists in an object.

```
let obj={a:45,b:90,c:23}

  if(  "a"  in  obj ){
       console.log("yes")
  }else{
      console.log("no")
   }
```

17.Check if a value exists in an object?

```
    let obj={a:45,b:90,c:23}
     let valuetocheck=100
     let exists=false;

  for(let key in obj){
       if(obj[key]===valuetocheck){
          exists=true
           break;
      }
```

```
    }
    console.log(exists)
```

17.Convert array of objects to a single object using a property as key.

```
let users=[{ id: 1, name: "Alice" },
          { id: 2, name: "Bob" },
          { id: 3, name: "Charlie" }]


let result={}

   for(let i=0;i<users.length;i++){
            let key=users[i].id;
            result[key]=users[i]
          }

    console.log(result)
```

JavaScript Concepts & Problems

Array Operations:

- [x] *Reverse an array* using for and while loops (without using an extra array).
- [x] *Push elements from one array to another* without using the push() method.
- [x] *Merge two arrays* using a for loop.
- [ ] *Swap two arrays* without using an extra array.
- [x] *Remove elements between two numbers* in an array.
- [x] *Find unique elements* in an array.
- [x] *Find duplicate elements* in an array.

- [x] *Find common elements* between two arrays.
- [x] *Sort an array* in ascending and descending order.

Functions & Logical Operations:

- [x] *Sum of two numbers using a callback function.*
- [x] *Function to check if a number is prime* using recursion.
- [x] *Generator function to print a multiplication table.*

Control Structures & Loops:

- [x] *Print even numbers from 1 to 100 in decreasing order* using a while loop.

JavaScript Fundamentals:

- [x] **Post-increment (i++) vs. Pre-increment (++i).**
- [x]
- [x] *Logical operators and expressions.*
- [x] *JavaScript operators (arithmetic, comparison, logical, etc.).*
- [x] *JavaScript keywords, identifiers, variables, and data types.*
- [x] **Difference between let, var, and const.**

Problem Solving:

- [x] *Find the sum of two numbers that equal a target value.*
- [x] *Find the greatest number among three variables.*
- [x] Write a program to merge two arrays using a for loop.
- [x] Swap two arrays without using an extra array.
- [x] Explain and demonstrate post-increment and pre-increment in JavaScript.
- [x] Find and print common elements from two arrays.
- [x] Explain the difference between let, var, and const in JavaScript.

- [x] Print even numbers from 100 to 1 in decreasing order using a while loop.

- [x] Given two arrays:

Arr1 = [1,2,3,4]

Arr2 = [5,6,7,8]

Swap all elements of Arr2 into Arr1 and all elements of Arr1 into Arr2 without using push or spread operators.

- [x] Print the multiplication table of 7 using a do-while loop.

- [x] Solve Leetcode problem 412.