ARRAY BASED QUESTIONS JAVASCRIPT

```
// MAX VALUE

// const arr = [1, 2,2, 3, 4, 4,5];
// const maxNumber = arr.reduce(function(prev,curr){
//  return   prev > curr ? prev : curr
// })

// function maxnumarr(arr){
//   var max = arr[0] ;

//   for(var i = 0 ; i<arr.length ; i++){
//     if(arr[i] > max){
//       max=arr[i]
//     }
//   }
//   return max

// }

// var arr=[12,1,78,90,45];

// console.log(maxnumarr(arr))

//MIN VALUE

// const minValue = arr.reduce(function(prev,curr){
//                   return prev>curr ? curr : prev
// })
// console.log(minValue)

//----------FIND UNIQUE VALUE OF ARRAY----------

// const uniqueVal = arr.filter((elem,index,arr) => arr.indexOf(elem) == index);

// console.log(uniqueVal)

//----------SORT THIS ARRAY------------------------------

// const unsortedArr = uniqueVal.sort(function(a,b) {
//                   return a-b
// })
```

```
// console.log(unsortedArr)

//---------------------FIND DUPLICATE element OF ARRAY--------------------
// function duplicateelem(arr) {
//   var count = 0;
//   var newarr = [];

//   for (let i = 0; i < arr.length; i++) {
//     let isDuplicate = false;

//     // Check if the element is already in newarr
//     for (let j = 0; j < newarr.length; j++) {
//       if (arr[i] === newarr[j]) {
//         isDuplicate = true;
//         break;
//       }
//     }

//     if (!isDuplicate) {
//       for (let j = i + 1; j < arr.length; j++) {
//         if (arr[i] === arr[j]) {
//           count++;
//           newarr.push(arr[i]);
//           break;
//         }
//       }
//     }
//   }

//   console.log(`There are ${count} given duplicate elements: ${newarr}`);
// }
//-------------------------REMOVE DUPLICATE USING FILTER-------------------------------

// const duplicateElem = arr.filter((elem,index,arr) => arr.indexOf(elem) !== index)
// console.log(duplicateElem)

// console.log(newarr);
// console.log("duplicate element : " + count)

//---------------AVERAGE NUMBER OF ELEMENTS IN ARRAY---------------------

// var arr = [1,2,3,4,5];
```

```
// function averagenum(arr){
//   var len = arr.length;
//   let total = 0;
// for(var i = 0 ; i<len ; i++){
//   total = total + arr[i]
// }

// var  average = total/len;
// return average;

// }
// console.log(averagenum(arr))

//------------------SPREAD OPERATOR ------------------------------

// let arr1 = [1, 23, 12];
// let arr2 = [23, 1, 56, 67];

// let concatearr = [...arr1, ...arr2];

// console.log(concatearr);

// function spreadOpr(a, b, c) {
//   console.log(a);
//   console.log(b);
//   console.log(c);
// }
// spreadOpr(...arr2)

//---------------OBJECT IN JAVASCRIPT--------------------
// const data = {
//   name: "sweta",

//   role: "34",
//   brach: {
//     computer: {
//       book: "c++",
//     },
//   },
// };

// console.log(data.brach);
```

```
//-------------THIS IN JS--------------------

// class c1 {
//   num = 24;
//   f1() {
//     console.log("this is the given number" + this.num);
//   }
//   f2() {
//     this.f1();
//   }
// }

//--------------TEMPLATE LITRALS-------------------

// var age= 20;
// var name = "sweta";

// console.log(`my name is ${name} my age is ${age}`)

//-----------------HOW TO CLONE A OBJECT IN JAVASCRIPT------------------
// const obj1 = {name:"sweta" , age:"25" };
// const obj2 = {job:"software developer"};

// const newObj = Object.assign(obj1 , obj2)
// console.log(newObj)

//-----------------ARRAY BASED QUESTIONS1-----------------------

// function foobar(N) {
//   var emptyArr = [];

//   for (var i = 1; i < N; i++) {
//     if (i % 3 === 0) {
//       emptyArr.push("FOO");
//     } else if (i % 5 === 0) {
//       emptyArr.push("BAR");
//     } else if (i % 3 === 0 && i % 5 === 0) {
//       emptyArr.push("FOOBAR");
//     } else {
//       emptyArr.push(i);
//     }
//   }
//   return emptyArr;
```

```
// }

// console.log(
//   JSON.stringify(foobar(15)) ===
//     JSON.stringify([
//       1,
//       2,
//       "Foo",
//       4,
//       "Bar",
//       "Foo",
//       7,
//       8,
//       "Foo",
//       "Bar",
//       11,
//       "Foo",
//       13,
//       14,
//       "FooBar",
//     ])
// );

//--------------------------DUPLICATE NUM --------------------

// var newArr = [12, 1, 23, 1];
// var count = 0;
// for (var i = 0; i <= newArr.length; i++) {
//   for (var j = i + 1; j <= newArr.length; j++) {
//     if (newArr[i] === newArr[j]) {
//       count++;
//       break;
//     }
//   }
//   if (count > 0) {
//     break;
//   }
// }
// const hasduplicate = count > 0;

// console.log("duplicate elements are : " + count + " so it is " + hasduplicate);

//----------------------CALL , APPLY AND BIND-------------------------
```

```
//CALL------------
// function sayHello(){
//  return console.log("hello " + this.name)
// }
// obj = {name:"sweta"}

// sayHello.call(obj)

// //APPLY-----------------
// function sendmsg (message) {
// return console.log(this.name + " is " + message)

// }
// var person = {name:"sweta"}

// sendmsg.apply(person,["good girl"])

// //BIND----------------------
// function sendMsg(message) {
//   return console.log(this.name + " is " + message);
// }

// var person = { name: "Sweta" };
// var boundSendMsg = sendMsg.bind(person);

// boundSendMsg("a good girl"); // Output: Sweta is a good girl


//----------------------curring in javascript---------------------------------
// function add(a) {
//   return function (b) {
//     return a + b;
//   };
// }

// console.log(add(3)(4));

//HOW TO FIND SECOND LARGEST VALUE OF ARRAY

// var arr = [1,23,1,34,12,45];
// arr.sort((a,b) => a-b);

// console.log("Largest Val : " +arr[(arr.length-1)])
```

```
// console.log("Second Largest Val : " +arr[(arr.length-2)])
// console.log("Smallest Val : " +arr[(arr[0])])
// console.log(arr)

//--------------------------FIND THE LENGTH OF ARRAY WITHOUT LENGTH METHOD
--------------------
// var arr =[12,2,4,2,1];

// function arrLen(arr){
// var length = 0;
// while(arr[length] != undefined){
//                   length++
// }
// return length;
// }
// console.log(arrLen(arr))

//-----------------CREATE A NEW ARRAY AND AND PRINT ALL THE ELEMENTS OF
ARRAY---------------

// var newArr = new Array();
// newArr[0] ="sweta";
// newArr[1] ="sskdmk";
// newArr[2] ="1";
// newArr[3] =12;

// for(var i = 0; i<newArr.length ; i++){
//                   console.log(newArr[i])
// }

//----------------FIND EVEN AND ODD ELEMENTS OF GIVEN ARRAY------------------

// var arr = [12, 34, 23, 12, 45, 12];
// function evenOdd(arr) {

//                   var evenArr = [];
//                   var oddArr =[];
//   for (var i = 0; i < arr.length; i++) {
//     if (arr[i] % 2 == 0) {

//       evenArr.push(arr[i])
//       console.log("this is even no Array = " + evenArr);
//     } else {
```

```
//       oddArr.push(arr[i])
//       console.log("this is odd number Array =" + oddArr);
//     }

//   }

// }
// var a = [12, 4, 23, 1, 5, 12];
// console.log(evenOdd(a))

//-----------------------REVERSE OF ARRAY WITHOUT USING REVERSE METHOD
----------------------------------------

// function reverse(arr) {
//   for (var i = arr.length -1; i >= 0; i--) {
// console.log(arr[i])
//   }
// }
// var a = [34,67,89,90,8];

// console.log(reverse(a))

//-------------------------HOW TO REMOVE SPECIFIC ELEMENT OF AN ARRAY
---------------------------------

//using indexof

// var arr = [12, 34, 45, 78, 1];
// const  i = arr.indexOf(1);

// if(i>-1){
// arr.splice(i,1)
// }
// console.log(arr)

//USING FILTER
// let value = 45;
// const removeElem = arr.filter(item => item !== value)
// console.log(removeElem)

// let count = 0;
```

```
//USING NORMAL METHOD
// function removeelem(arr,val){
//   let count = 0
//   var newarr  = [];
//   for(var i = 0;i<arr.length ;i++){
//     if(arr[i] === val && count === 0){
//       count++;

//     }else{
//       newarr.push(arr[i])
//     }
//   }
//   return newarr
// }
// console.log(removeelem(arr,4))

// ----------------------HOW TO FIND THE MISSING ELEMENT OF
ARRAY-----------------------------------------------

// function missingElem(arr) {
//   var n = arr.length + 1;
//   for (var i = 1; i < n; i++) {
//     if (!arr.includes(i)) {
//       return i;
//     }
//   }
//   return n;
// }
// var arr = [1, 2, 3, 4, 5];
// console.log(missingElem(a))

//--------------------------------SUM OF ALL ELEMENT OF ARRAY----------------------

// function SumOFNum(a) {
//   var sum = 0;
//   for (var i = 0; i < arr.length; i++) {
//     sum += a[i];
//   }
//   return sum;
// }
// console.log(SumOFNum(arr))

//----------------------FIND FACTORIAL OF GIVEN NUMBER-----------------
```

```
// function FctorialNum(n) {

//   if (n === 0 || n === 1) {
//     return 1;
//   } else {
//     return n * FctorialNum(n - 1);
//   }
// }
// console.log(FctorialNum(1))
//-------------------------PRIME NUMBER-----------------------------------------

// function isPrime(number) {
//   if (number <= 1) {
//     return false;
//   }

//   for (let i = 2; i <= Math.sqrt(n); i++) {
//     if (n % i === 0) {
//       return false;
//     }
//   }
//   return true;
// }
// console.log(isPrime(n));

//-----------------------PALLINDROM ----------------------------------

// function palindromFun(str) {
//   str = str.toLowerCase();
//   const reversedStr = str.split("").reverse().join("");

//   if (reversedStr === str) {
//     return true;
//   } else {
//     return false;
//   }
// }

// console.log(palindromFun("sweta"));

//-----------------------------------FIZZBUZZZ-----------------------------------------
```

```
// function fizzBuzz(N) {
//   for (var i = 1; i <= N; i++) {
//     if (i % 3 === 0 && i % 5 === 0) {
//       console.log("fizzBuzz");
//     } else if (i % 5 === 0) {
//       console.log("buzz");
//     } else if (i % 3 === 0) {
//       console.log("fizz");
//     } else {
//       console.log(i);
//     }
//   }
// }
// const output = fizzBuzz(20);
// console.log(output);

//-----------------------CURRING FUNCTION-----------------------------------

function mul(x) {
  return function (y) {
    return function (z) {
      return x * y * z;
    };
  };
}
// console.log(mul(2)(3)(4))

// function muldk(x,y,z){
//                 return x*y*z;
// }
// const output = muldk(2,3,5);
// console.log(output)

//-----------------------EMPTY A ARRAY-------------------------------

// var arrayList = ['a','d','d','e'];
// var newArrayList = arrayList;
//  arrayList = [];
//  console.log(arrayList);

//-------------------REVERSE A STRING ARRAY-------------------------------
// function revstrarr(s) {
//   var left = 0;
```

```
//   var right = s.length - 1;

//   while (left < right) {
//     const temp = s[left];
//     s[left] = s[right];
//     s[right] = temp;

//     left++;
//     right--;
//   }

//   return s;
// }

// const a = ["s", "w", "e"];
// console.log(revstrarr(a));

//-----------------------------------REMOVE OBJECT FROM AN
ARRAY-------------------------------------------

// let arr2 = [
//   {name:"sweta"},
//   {name:"aditi"},
//   {name:"nisha"}
// ]

// let objtoremove = "aditi";

// for(var i = 0 ; i<arr2.length ; i++){
//   if(arr2[i].name === objtoremove){
//     arr2.splice(i,1)
//     break;
//   }
// }
// console.log(arr2)

//----------------------INTERSECTION USING
FILTER-----------------------------------------------------------------------

// function intersectionarr(arr1,arr2){

//   return arr1.filter(item => arr2.includes(item))
// }
```

```
// console.log(intersectionarr(a,b))

//---Write a function that takes an array of numbers as input and returns the sum of all the positive numbers in the array.----------

// function sumarr(arr){

//   var sum = 0;
//   for(var i = 0 ; i<arr.length ; i++){
//     if(arr[i] >= 1 ){
// sum += arr[i]
//     }

//   }
//   return sum

// }
// const ar = [1,2,0,-1]
// console.log(sumarr(ar))

//Given an array of integers, write a function that finds the two numbers that add up to a given target number. Return the indices of the two numbers in the array.----------

// function returnsum(arr,target){

//   for(let  i = 0 ;i <= arr.length ; i++){
//     for(j=i+1 ; j<=arr.length ; j++){
//       if(arr[i] + arr[j] == target){
//         return [i,j]
//       }
//     }
//   }
//   return null

// }

// const arr1 =[2, 7, 11, 15]
// const targetsum = 18
// console.log(returnsum(arr1,targetsum))

//----------FIND THE SECOND LARGEST NUMBER OF ARRAY----------

// function secondLarge(arr){
```

```
//   var max = -Infinity;
//   var secondmax = -Infinity;
//   for(var i = 0; i<arr.length ; i++){
//     if(arr[i] > max){
//       secondmax = max;
//       max=arr[i]
//     }else if(arr[i] > secondmax && arr[i] !== max){
//       secondmax = arr[i]
//     }

//   }
//   return secondmax
// }
// const ar = [1, 5, 2, 9, 6,9];
// const secondLargest = secondLarge(ar);
// console.log(secondLargest);

//--------------------------UNION OF ARRAY---------------------------------
// function unionofarr(arr1,arr2){
//   const combinedarr = arr1.concat(arr2)
// const unionset = new Set(combinedarr);
// const unionArray = Array.from(unionset);
// return unionArray

// }
// const array1 = [1, 2, 3, 4];
// const array2 = [3, 4, 5, 6];
// const unionResult = unionofarr(array1, array2);
// console.log(unionResult);

//----------------------------SORT AN ARRAY WITHOUT USING SORT
METHOD-------------------------

// function sortarr(arr){
//   var len = arr.length;
//   for(var i = 0; i<len ; i++){
//     for(var j = 0 ; j<len-1; j++)
//     {
//       if(arr[j] > arr[j+1]){
//         let temp = arr[j];
//         arr[j] = arr[j+1];
//         arr[j+1] = temp
//       }
```

```
//     }
//   }
//   return arr
// }

// var c=[23,1,34,12]
// console.log(sortarr(c))

//PALLINDROM NUMBER-----------------------------------------------

// function ispallindrome(num){
//   if(num<0){

//     return false
//   }

//   let originalnum = num;
//   let reversenum = 0 ;

//  while(num>0){
//   let lastdigit = num%10;
//   reversenum = reversenum*10 + lastdigit;
//   num = Math.floor(num/10)

//  }
//  return originalnum === reversenum
// }

// console.log(ispallindrome(1122))

//--------------------------------ARMSTRONG NUMBER----------------------------------------

// function armstronnum(num){
//   const numstr = num.toString();
//   const numdigit = numstr.length;
//   let sum = 0;

//   for(var i = 0; i<numdigit ; i++){
//     const digit = parseInt(numstr[i]);
//     sum += Math.pow(digit , numdigit);
//   }
//   return sum === num
```

```
// }

// console.log(armstronnum(153))

//write a function that will take an array and give the number of count that how many time the number is paresent in array----------
//eg.--arr=['a','b','a'] ---op = a-2,b-1

// function numofarr(arr)
// {
//   const count = {};

//   for(const item of arr ){
//     if(count[item] ){
//       count[item]++
//     }else{
//       count[item] = 1
//     }
//   }

// return count;
// }

// const arr = [1,2,1,3,3,4,3]
// const result = numofarr(arr)

// for(const key in result){
//   console.log(`${key} - ${result[key]}`)
// }

// Implement a function that returns an updated array with r right rotations on an array of integers a .--------------------------

// function rightroration(arr,rotations){
//   if(rotations<0){
//     return false
//   }
// for(let i = 0; i<rotations;i++){
//   let elements = arr.pop();
//   arr.unshift(elements);

// }
// return arr
```

```
// }
// var arr = [2, 3, 4, 5, 7]
// console.log(rightroration(arr ,3))

// function leftRotation(arr,rotation){
//   for(let i =0;i<rotation ; i++){
//     let first = arr.shift();
//     arr.push(first)

//   }
//   return arr
// }

//---------------------------------------ARRAY DESTRUCTURING-----------------------------------------------

//we can use spread operator
// swapping two numbers without using third variable
// we can ingore variables using ,

//-------------es5

// const myproglang = ['js','java','c','c++','python']

// var top1 = myproglang[0];
// var top2 = myproglang[1];

// console.log("my fevrt prog lang is : " +top1)

//------------es6

// let [top1,top2,top3,top4,top5] = myproglang;

// console.log(`my fevrt programming lang is : ${top1}`)

//IF WE WANT LAST ELEMNENT ONLY THAN WE CAN DO IT LIKE THIS

// let[top1,,,,top5] = myproglang;
// console.log(`my fevrate prog lang is ${top5}`)

//program to swap two numbers without using third variable

// THIS IS USING THIRD VARIABLES
// let a = 5;
```

```
// let b = 6;

// let temp = a;
// a=b;
// b=temp;

// console.log( `the value of a is ${a} and b is ${b}`)

//WITHOUT USING THIRD VARIABLES

// [a,b] = [b,a]

// console.log( `the value of a is ${a} and b is ${b}`)

//Write a function that takes an array of numbers as input and returns an object that
//contains the count of each number in the array.(VVI)

// function countnum(arr){

//   const count = {};

//   for(const item of arr){
//     if(count[item]){
//       count[item]++
//     }else{
//       count[item] = 1
//     }
//   }
//   return count
// }

// var arr = [1, 2, 3, 2, 1, 3, 3, 4, 5, 4];

//Write a function that takes an array of integers as input and returns the number that
//appears the most frequently in the array. If there is a tie, return any of the tied numbers.(VVI)

// function mostfreqnuminarr(arr){
//   const numCount = new Map();
//   let maxCount =0;
//   let mostfeqnum;
//   for(const num of arr){
//     numCount.set(num , (numCount.get(num) || 0) + 1);
//     if(numCount.get(num)>maxCount){
```

```
//       maxCount=numCount.set(num)
//       mostfeqnum = num
//     }

//   }

//   return mostfeqnum
// }
// const arr = [4, 2, 4, 3, 1, 4, 2, 2, 3]
// console.log(mostfreqnuminarr(arr))

// Implement a function to flatten a nested array. The function should take an array that may
contain other
// arrays of integers and return a single array with all the integers.(VVI)

// function flattenarray(nestedArr){
//   return nestedArr.reduce((flatenned,current) => {
//     if(Array.isArray(current)){
//       return flatenned.concat(flattenarray(current))
//     }else{
//       return flatenned.concat(current)
//     }

//   },[])
// }

const nestedArray = [1, [2, 3], [4, [5, 6]]];
// METHOD2
// const flatennedArray = nestedArray.flat(Infinity);
// console.log(flatennedArray)

// const a = [null , ,null]
// console.log(a[1])

//HOW TO ADD TWO ARRAY ELEMENT----------------------------------------------

// const array1 = [1, 2, 3];
// const array2 = [3, 4, 5, 6];
// const sum = [];

// const maxLength = Math.max(array1.length, array2.length);

// for (var i = 0; i < maxLength; i++) {
```

```
//   const element1 = array1[i] || 0;
//   const element2 = array2[i] || 0;

//   sum.push(element1 + element2);
// }
// console.log(sum);

//WRITE A PROGRAM TO MULTIPLY TWO NUMBER WITHOUT USING MULTIPLY SIGN

//WRITE A PROGRAM TO MAKE POLLYFILLS OF MAP ,FILTER AND REDUCE METHOD


STRING BASED QUESTIONS

//----------------------------------STRING BASED QUESTION IN
JAVASCRIPT-----------------------------------------------

// let str = "Hello I am Sweta";

//CONVERT INTO ARRAY

// console.log(str.split(' '))
// console.log([...str])

//REPLACE

// console.log(str.replace('Hello',"Hii"))
// console.log(str.replace(/a/g, "aaaa"))

//SUBSTRING

// console.log(str.substring(6,10))

//-----------------------------------------------STRING
PALLINDROME----------------------------------------------------

function ispallindrom(str) {
   str = str.toLowerCase().replace(/[^a-zA-Z0-9]/g, "");

  for (let i = 0; i < Math.floor(str.length / 2); i++) {
    if (str[i] !== str[str.length - 1 - i]) {
      return false;
    }
```

```
  }
  return true;
}
console.log(ispallindrom("A man, a plan, a canal, Panama"))

//-------------------------------------------------STRING REVERSE
------------------------------------------------------------

// function reverseStr(str) {
//   str = str.toLowerCase();

//    const reverseStr = str.split("").reverse().join("");
//   return reverseStr;
// }
// console.log(reverseStr("sweta"));

//----------------------------------------REVERSE THE ARRAY OF
STRING-----------------------------------------------

// function revstrarr(arr) {
//   var left = 0;
//   var right = arr.length - 1;

//   while(left<right){
//                   var temp = arr[left];
//                   arr[left] = arr[right];
//                   arr[right] = temp;
//   }
//   return arr

// }
// const a=["a","b","c"];
// console.log(revstrarr(a))

//-----------------------------FIND VOWELS FROM YOUR NAME
-----------------------------------------------------------------

// const name = "sweta";

// const vowel = name.match(/[aeiouAOU]/g);

// console.log(vowel)
```

```
//-----------------------------------------FUNCTION TO FIND THE VOWEL FROM YOUR NAME--------------------------------------

// function findVowel(str) {
//   const vowel = "aeiouAEIOU";
//   const vowelsare = [];

//   for (let i = 0; i < str.length; i++) {
//     if (vowel.includes(str[i])) {
//                     vowelsare.push(str[i])

//     }
//   }
//   return vowelsare;
// }
// const myname ="nisha";

// console.log(findVowel(myname))

//----------------------Implement a function that reverses a string without using the built-in reverse() function.----------

// function reversestr(str) {
//   let reverse = "";

//   for (let i = str.length - 1; i >= 0; i--) {
//     reverse += str[i];
//   }
//   return reverse
// }
// const string = "sweta";
// console.log("original array : " +reversestr(string))

//Create a function that converts a string to an array of characters----------

// function strtoarr(str) {
//   return Array.from(str);
// }
// var str = "I am sweta";
// console.log(strtoarr(str));
```

```
//WRITE A FUNCTION THAT WILL CONVERT ALL THE FIRST LETTTER OF STRING IN CAPITAL LETTER----------

//METHOD-1

// function capitalStr(sentance){
//   const words = sentance.split(' ');
//   const capitalWords = words.map(word => {
//     if(word.length > 0){
//       return word[0].toUpperCase() + word.slice(1).toLowerCase()
//     }else{
//       return ' ';
//     }
//   })
//   return capitalWords.join(' ')
// }

// var sentance = "hi I am sweta karn"
// console.log(capitalStr(sentance))

//METHOD-2

// function capfirstltr(str){
//   return str.replace(/\b\w/g,c=>c.toUpperCase())
// }
// console.log(capfirstltr("sweta"))

//STRING ANAGRAM ----------------------------------

// function stranagram(str1,str2){
//   //remove spaces and convert into lower case
//   str1 = str1.replace(/\s+/g, '').toLowerCase();
//   str2 = str2.replace(/\s+/g,'').toLowerCase();

//   const sortedStr1 = str1.split('').sort().join('');
//   const sortedStr2 = str2.split('').sort().join('');

//   return sortedStr1 === sortedStr2


// }
// const string1 = "listenw" ;
// const string2 = "silent";
```

```
// if (stranagram(string1, string2)) {
//   console.log("The strings are anagrams.");
// } else {
//   console.log("The strings are not anagrams.");
// }

//FIND THE LENGTH OF ARRAY WITHOUT LENGTH METHOD----------------------------------------

// function strlen(str){
//   var len = 0;


//   while(str[len] !== undefined){
//     len++
//   }
//   return len
// }
// console.log(strlen("sweta"))

//Write a function that takes two arrays of strings as input and returns a new array
// containing the common elements from both arrays, without any duplicates.

//string intersection

// function strintersection(str1 ,str2){
//   return str1.filter(item => str2.includes(item))
// }
// const arr1 = ["apple", "banana", "orange", "grape"];
// const arr2 = ["orange", "kiwi", "banana", "pineapple"];



//string union

// function unionostr(str1,str2){
//   const concatestr = str1.concat(str2);

//   const removeduplicate = new Set(concatestr);

//   const convertsetintoarray = Array.from(removeduplicate);
//    return convertsetintoarray
// }
```

```
// //Implement a function that takes a string as input and returns the most frequent character(s) in the string.
// //If there is a tie, return all the characters that are tied.(VVI)

// function charchaountstr(str){
//   const charCount = new Map();

//   let maxCount = 0;

//   for(const char of str){
//     if(char !== ' '){

//       charCount.set(char,(charCount.get(char)|| 0) + 1);
//       maxCount = Math.max(maxCount , charCount.get(char))

//     }
//   }
//   const mostFreqchar = [];
//   for(const [char,count] of charCount)
//   {
//     if(count === maxCount){
//       mostFreqchar.push(char)
//     }

//   }
//   return mostFreqchar
// }

// const str = "hello word";

// console.log(charchaountstr(str))

//Write a function that takes a map as input and returns an array of the keys sorted in descending order based on their values.
//For example, given the input: { "apple": 2, "banana": 4, "orange": 1 }, the function should return: ["banana", "apple", "orange"]./


// function sortkeysbasedonvalue(inputmap){
//   const sortedkeys = Object.keys(inputmap).sort((a,b) => inputmap[b]-inputmap[a])
//   return sortedkeys
// }
```

```
// const inputMap = { "apple": 2, "banana": 4, "orange": 1 };

// console.log(sortkeysbasedonvalue(inputMap))
```

PROMISE BASED QUESTIONS JAVASCRIPT

```
//-----------------------------------------------PROMISE-----------------------------------------------------------

// function sumofnums(...element) {
//   return new Promise((resolve, reject) => {
//     if (element.length > 3) {
//       reject("Only three elements or less are allowed");
//     } else {
//       let sum = 0;
//       let i = 0;
//       while (i < element.length) {
//         sum += element[i];
//         i++;
//       }
//       resolve("sum of elements is : " +sum)
//     }

//   });
// }

// // console.log(sumofnums(1,2,3))
// sumofnums(1,2,3,4)
// .then(result => console.log(result))
// .catch(error => console.log(error))

//-----------------------------------------ASYNC/AWAIT-----------------------------------------

async function sumofnums(...elements){
  return new Promise((resolve , reject) => {
    if(elements.length>3){
      reject('element more than 3 is not allowed')
    }
    else{
      let sum = 0;
```

```
      let i=0;
      while(i<elements.length){
        sum +=elements[i];
        i++;
      }
      resolve('the sum of elements is:' +sum)
    }
  })
}

async function calculatesum(){
  try{
    const result = await sumofnums(1,2,3,5);
    console.log(result)

  }catch(error){
    console.log(error)
  }
}
calculatesum()



//-----------------NUMBERS-----------------------------

//multiplication table from 1 TO 10-----------------------------------------

// for(var i = 1 ; i<=10 ; i++){
//   console.log(`Multiplication table of ${i}`)
//   for(var j = 1 ; j<=10 ; j++){
//     var result = i * j;
//     console.log(`${i} * ${j} = ${result}`)

//   }
//   console.log(" ")
// }

//Create a function that will convert from Celsius to Fahrenheit--------------------

// function CelsiustoFarenheight(n){
//  return n*1.8+32

// }
```

```
// let r = 20;
// console.log(CelsiustoFarenheight(r))

// function FarenheighttoCelcious(n){
//                     return (n-32)/1.8

// }
// console.log(FarenheighttoCelcious(r))

//---Print the first 10 Fibonacci numbers without recursion----------------------------------------
// let f0 = 0;
// let f1 = 1;

// for (var i = 2; i < 10; i++) {
//   let fi = f0 + f1;

//   f0 = f1;
//   f1 = fi;
//   console.log(fi);
// }

//Create a function that will find the nth Fibonacci number using recursion----------

// function FibonacciNum(n) {
//   if (n == 0) {
//     return 0;
//   }
//    if (n == 1) {
//     return 1;
//   }

//   return FibonacciNum(n-1)+FibonacciNum(n-2)

// }
// let n = 10
// console.log(FibonacciNum(n))

//--------------------------------Rotate an array to the left 1 position

// function rotateLeft(arr) {
//   let first = arr.shift();
//   arr.push(first);
// }
```

```
// var ar = [1, 3, 4];
// // rotateLeft(ar);
// // console.log(ar);

// function rotatetoright(arr) {
//   let last = arr.pop();
//   arr.unshift(last);
// }

// rotatetoright(ar)
// console.log(ar)

//----------Calculate the sum of first 100 prime numbers
```

METHODS

```
//----------SLICE----------

// it is used to create a new array each time contain the portion of ariginal array
//it takes two argument
// 1.starting index(inclusive)
// 2.ending index (exclusive)

//it will return the new array

// const originalarr = [1, 2, 3, 4, 5];
// const newarr = originalarr.slice(2, 4); //here 1 and 5 will be removed and we will gwt an array
of [2,3,4]
// console.log(newarr);
// console.log(originalarr)

//----------SPLICE----------

// it will not create the new array it will modify the existing array
// it will take --

// 1 .first index
// 2.the number of Element to remove
// 3. optional Element to insert at the location

// const arr = [1, 2, 3, 4, 5];
```

```
// const removeelem = arr.splice(0, 1);
// console.log(removeelem)

//---------------------------------------REST PARAMETER------------------------------

// function addnums(...nums) {
//   let sum = 0;
//   let i = 0;
//   while (i < nums.length) {
//     sum += nums[i];
//     i++;
//   }
//   return sum;
// }

// console.log(addnums(1,2,3,3,4))

//CHARAT()

//-------------------THIS KEYWORD----------------------

//WHEN THERE IS NOTHING WITH THIS KEYWORD THAN IT WILL REFFER TO THE
WINDOW OBJECT

// console.log(this);

//THIS KEYWORD WITH OBJECT---------------

// const obj = {
//   firstName: "sweta",
//   lastname: "karn",
//   fullName: function() {
//     return this.firstName + " " + this.lastname;
//   },
// };

// console.log(obj.fullName())

//THIS KEYWORD WITH FUNCTION

// "use strict"
// function thiskey(){
// return this;
```

```
// }
// console.log(thiskey())
//---------------------------------------------------------------- String Methods:
// charAt(index)
// concat(str1, str2)
// indexOf(searchValue, startIndex)
// toLowerCase()
// toUpperCase()
// trim()
// replace(searchValue, replaceValue)
// substring(startIndex, endIndex)
// split(separator)
// length
//---------------------------------- ----------------------------Array Methods:
// push(element)
// pop()
// shift()
// unshift(element)
// concat(array1, array2)
// join(separator)
// indexOf(searchElement, startIndex)
// splice(startIndex, deleteCount, element1, element2, ...)
// slice(startIndex, endIndex)
// forEach(callback)
// map(callback)
// filter(callback)
// reduce(callback, initialValue)
// ----------------------------------------------------------------Object Methods:
// Object.keys(obj)
// Object.values(obj)
// Object.entries(obj)
// hasOwnProperty(property)
// assign(target, source1, source2, ...)
// ----------------------------------------------------------------Number Methods:
// toFixed(decimalPlaces)
// toPrecision(precision)
// parseInt(string, radix)
// parseFloat(string)
```