**1) find pair of array**

var a = [3,2,10,5,4,9,12,16,7,0,6]

let count = 0;

for(let i=0; i<=a.length; i++){

for(let j = i + 1; j<=a.length;j++){

if(a[i] + a[j] == 10){

console.log("yess");

count++;

}

}

}

console.log(count);

output = 3

**2) some of array using reduce method**

const numbers = [1, 2, 3, 4, 5];

const sum = numbers.reduce((accumulator, currentValue) => {

return accumulator + currentValue;

}, 0);

**3) let str = "Vikram chauhan developer";**

// Convert to lowercase and split into words

let words = str.toLowerCase().split(' ');

console.log("lowercase", words);

// Capitalize the first letter of each word (except the first word)

for (let i = 2; i < words.length; i++) {

words[i] = words[i].charAt(0).toUpperCase() + words[i].slice(1);

}

console.log("join", words);

// Join the words back together

let result = words.join('');

console.log(result);

output = vikramchauhanDeveloper

**4) How to convert object to array in javacsript**

const obj = {

a: 'hello',

b: 'hi'

}

// const arr = [

// [

// "a",

// "hello"

// ],

// [

// "b",

// "hi"

// ]

// ]

const arr = Object.entries(obj);

const objj = {};

arr.forEach(([key, value]) => {

objj[key] = value;

});

console.log(obj);

**5) const numbers = [10,20,30];**

let sum = 0;

for(let i = 0; i < numbers.length; i++){

sum += numbers[i];

}

console.log(sum)

=================================================================================

**6) find the occurrence of name**

const user = ["alex","bob","alex","cary"];

const counts = {};

for(let i = 0; i < user.length; i++){

const name = user[i].replace(/'/g, ' ');

counts[name] = (counts[name] || 0) + 1;

}

const res = Object.keys(counts).map(name => ({

name,

count: counts[name]

}))

console.log(res);

Output

[

{

name: "alex",

count: 2

},

{

name: "bob",

count: 1

}

{

name: "cary",

count: 1

}

]

==================================================================================

**7) how can I find sum of number in string**

const str = "VI1k5r7a6m9si4nh3";

const str = "VI1k5r7a6m9si4nh3";

function sumOfNumbersInString(inputString) {

let sum = 0;

for (let i = 0; i < inputString.length; i++) {

const currentChar = inputString[i];

// Check if the character is a digit

if (!isNaN(parseInt(currentChar))) {

// If it's a digit, add it to the sum

sum += parseInt(currentChar);

}

}

return sum;

}

const result = sumOfNumbersInString(str);

console.log(result); // Output: 35

for(let i = 0; i < str.length; i++){

if(!isNaN(parseInt(str[i]))){

sum += parseInt(str[i])

}

}

==================================================================================

**8) sum of this number**

const str = {"1.2","25.5","35.6","47.8","65.5","78.5"}

// Initialize sum to 0

let sum = 0;

// Loop through the array and add each element to the sum

for (let i = 0; i < str.length; i++) {

// Convert the string to a number using parseFloat

const num = parseFloat(str[i]);

// Check if the conversion is successful (not NaN)

if (!isNaN(num)) {

sum += num;

}

}

console.log(sum);

---------------------------------------------------

using reduce method

const str = ["1.2", "25.5", "35.6", "47.8", "65.5", "78.5"];

// Convert strings to numbers and calculate the sum

const sum = str.reduce((acc, num) => acc + parseFloat(num), 0);

console.log(sum);

==================================================================================

**9) find the unique from array** const arr = [1, 1, 2, 3, 4, 4, 4, 5, 5, 6];

**Method 1: Using a loop**

const uniqueArray = [];

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

if (uniqueArray.indexOf(arr[i]) === -1) {

uniqueArray.push(arr[i]);

}

}

console.log(uniqueArray);

**Method 3: Using Set**

const uniqueSet = Array.from(new Set(arr));

console.log(uniqueSet);

**using filter**

const uniqueFilter = arr.filter((value, index, self) => self.indexOf(value) === index);

console.log(uniqueFilter);

**10) find duplicate value from array**

**Method 1: Using Nested Loops**

const arr = [1, 1, 2, 3, 4, 4, 4, 5, 5, 6];

const duplicates = [];

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

for (let j = i + 1; j < arr.length; j++) {

if (arr[i] === arr[j] && !duplicates.includes(arr[i])) {

duplicates.push(arr[i]);

}

}

}

console.log(duplicates);

**Method 2: Using an Object to Track Occurrences**

const arr = [1, 1, 2, 3, 4, 4, 4, 5, 5, 6];

const occurrences = {};

const duplicatesObject = [];

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

const num = arr[i];

occurrences[num] = (occurrences[num] || 0) + 1;

if (occurrences[num] === 2) {

duplicatesObject.push(num);

}

}

console.log(duplicatesObject);

**using filter**

const duplicatesSet = Array.from(new Set(arr.filter((value, index, self) => self.indexOf(value) !== index)));

console.log(duplicatesSet);

**using set**

const duplicatesSet = [...new Set(arr.filter((value, index, self) => self.indexOf(value) !== index))];

console.log(duplicatesSet);

**11) [0,5,1,3,2,3,4,1,6,0] put duplicate elements on right side of the array**

const arr = [0, 5, 1, 3, 2, 3, 4, 1, 6, 0];

const duplicatesOnRight = [];

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

const currentElement = arr[i];

// If the element is not in duplicatesOnRight, push it to duplicatesOnRight

if (!duplicatesOnRight.includes(currentElement)) {

duplicatesOnRight.push(currentElement);

} else {

// If the element is already in duplicatesOnRight, remove it from its current position

const index = arr.indexOf(currentElement, i);

arr.splice(index, 1);

// Add the element to duplicatesOnRight

duplicatesOnRight.push(currentElement);

}

}

console.log(arr.concat(duplicatesOnRight));

**12) [0,[1,2],[3,4,[5],6]] convert this to [0,1,2,3,4,5,6]**

**1) using nested arry**

const nestedArray = [0, [1, 2], [3, 4, [5], 6]];

function flattenArray(arr) {

let result = [];

arr.forEach((element) => {

if (Array.isArray(element)) {

result = result.concat(flattenArray(element));

} else {

result.push(element);

}

});

return result;

}

const flattenedArray = flattenArray(nestedArray);

console.log(flattenedArray);

**2) using reduce method**

const nestedArray = [0, [1, 2], [3, 4, [5], 6]];

function flattenArray(arr) {

return arr.reduce((result, element) => {

return result.concat(Array.isArray(element) ? flattenArray(element) : element);

}, []);

}

const flattenedArray = flattenArray(nestedArray);

console.log(flattenedArray);

**13) const arr = [1,1,2,5,3,2,4,1,6,5,7,3];**

find and count the occurance of the array and output display in array

const arr = [1, 1, 2, 5, 3, 2, 4, 1, 6, 5, 7, 3];

// Function to count occurrences and create a result array

const countOccurrences = (array) => {

const occurrences = {};

// Count occurrences

array.forEach((element) => {

occurrences[element] = (occurrences[element] || 0) + 1;

});

// Create result array

const resultArray = Object.keys(occurrences).map((key) => ({

element: parseInt(key),

count: occurrences[key],

}));

return resultArray;

};

// Get the result array

const result = countOccurrences(arr);

// Display the result

console.log(result);

output:

[

{ element: 1, count: 3 },

{ element: 2, count: 2 },

{ element: 5, count: 2 },

{ element: 3, count: 2 },

{ element: 4, count: 1 },

{ element: 6, count: 1 },

{ element: 7, count: 1 }

]

-----------------------------------------------------------------------------------------------

const arr = [1, 1, 2, 5, 3, 2, 4, 1, 6, 5, 7, 3];

const countOccurrences = (array) => {

const occurrences = {};

array.forEach((element) => {

occurrences[element] = (occurrences[element] || 0) + 1;

});

return occurrences;

};

const resultObject = countOccurrences(arr);

console.log(resultObject);

output:

{ '1': 3, '2': 2, '3': 2, '4': 1, '5': 2, '6': 1, '7': 1 }

-------------------------------------------------------------------------------------------------

**output as a string**

const arr = [1, 1, 2, 5, 3, 2, 4, 1, 6, 5, 7, 3];

const countOccurrences = (array) => {

const occurrences = {};

array.forEach((element) => {

occurrences[element] = (occurrences[element] || 0) + 1;

});

return Object.keys(occurrences)

.map((key) => `${key}: ${occurrences[key]}`)

.join(', ');

};

const resultString = countOccurrences(arr);

console.log(resultString);

**14) const str = "my name is vikram";**

// Split the string into an array of words

const wordsArray = str.split(' ');

console.log(wordsArray)

// Reverse the array of words

const reversedArray = wordsArray.reverse();

console.log(reversedArray)

// Join the reversed array back into a string

const reversedString = reversedArray.join(' ');

console.log(reversedString);

output: vikram is name my