**Arrays**

**1.** Write a JavaScript program to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number

Input: numbers= [10,20,10,40,50,60,70], target=50  
Output: 3, 4

function twoSum(nums, target\_num) {

var map = [];

var indexnum = [];

for (var x = 0; x < nums.length; x++){

if (map[nums[x]] != null){

var index = map[nums[x]];

indexnum[0] = index+1;

indexnum[1] = x+1;

break;

}

else{

map[target\_num - nums[x]] = x;

}

}

return indexnum;

}

console.log(twoSum([10,20,10,40,50,60,70],50));

**Output:**[3,4]

**2.** Write a JavaScript function to generate an array between two integers of 1 step length.

Test Data :  
console.log(rangeBetwee(4, 7));   
[4, 5, 6, 7]  
console.log(rangeBetwee(-4, 7));  
[-4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7]

function rangeBetwee(start, end) {

if (start > end) {

var arr = new Array(start - end + 1);

for (var i = 0; i < arr.length; i++, start--) {

resarrult[i] = start;

}

return arr;

}

else {

var arro = new Array(end-start+1);

for (var j = 0; j < arro.length; j++, start++) {

arro[j] = start;

}

return arro;

}

}

console.log(rangeBetwee(4, 7));

console.log(rangeBetwee(-4, 7));

**Output:**

[4,5,6,7]

[-4,-3,-2,-1,0,1,2,3,4,5,6,7]

**3.** Write a JavaScript program which prints the elements of the following array.    
Note : Use nested for loops.  
Sample array : var a = [[1, 2, 1, 24], [8, 11, 9, 4], [7, 0, 7, 27], [7, 4, 28, 14], [3, 10, 26, 7]];  
*Sample Output* :   
"row 0"   
" 1"   
" 2"   
" 1"  
" 24"

"row 1"   
------  
------

var a = [[1, 2, 1, 24], [8, 11, 9, 4], [7, 0, 7, 27], [7, 4, 28, 14], [3, 10, 26, 7]];

for (var i in a) {

console.log("row " + i);

for (var j in a[i]) {

console.log(" " + a[i][j]);

}

}

**Output:**

row 0

1

2

1

24

row 1

8

11

9

4

row 2

7

0

7

21

row 3

7

4

28

14

row 4

3

10

26

7

**4.** Write a JavaScript program which accept a number as input and insert dashes (-) between each two even numbers. For example if you accept 025468 the output should be 0-254-6-8.

var num = window.prompt();

var str = num.toString();

var result = [str[0]];

for (var x = 1; x < str.length; x++) {

if (str[x - 1] % 2 === 0 && str[x] % 2 === 0) {

result.push('-', str[x]);

} else {

result.push(str[x]);

}

}

console.log(result.join(''));

**6**. Write a JavaScript function to clone an array   
Test Data :  
console.log(array\_Clone([1, 2, 4, 0]));   
console.log(array\_Clone([1, 2, [4, 0]]));  
[1, 2, 4, 0]   
[1, 2, [4, 0]]

var array\_Clone = function(arra1) {

return arra1.slice(0);

};

console.log(array\_Clone([1, 2, 4, 0]));

console.log(array\_Clone([1, 2, [4, 0]]));

**Output:**

[1, 2, 4, 0]

[1, 2, [4, 0]]

**7.** Write a JavaScript program to flatten a nested (any depth) array. If you pass shallow, the array will only be flattened a single level.

Sample Data :  
console.log(flatten([1, [2], [3, [[4]]],[5,6]]));   
[1, 2, 3, 4, 5, 6]  
console.log(flatten([1, [2], [3, [[4]]],[5,6]], true));   
[1, 2, 3, [[4]], 5, 6]

var flatten = function(a, shallow, r){

if(!r){ r = []}

if (shallow) {

return r.concat.apply(r,a);

}

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

if(a[i].constructor == Array){

flatten(a[i], shallow, r);

}else{

r.push(a[i]);

}

}

return r;

}

console.log(flatten([1, [2], [3, [[4]]],[5,6]]));

console.log(flatten([1, [2], [3, [[4]]],[5,6]], true));

Output:

[1, 2, 3, 4, 5, 6]

[1, 2, 3,[[4]], 5, 6]

**8.** Write a JavaScript function to generate an array of specified length, filled with integer numbers, increase by one from starting position.

Test Data :  
console.log(array\_range(1, 4));   
[1, 2, 3, 4]  
console.log(array\_range(-6, 4));  
[-6, -5, -4, -3]

function array\_range(start, len)

{

        var arr = new Array(len);

        for (var i = 0; i < len; i++, start++)

{

            arr[i] = start;

        }

      return arr;

}

console.log(array\_range(1, 4));

console.log(array\_range(-6, 4));

**Output:**

[1, 2, 3, 4]

[-6, -5, -4, -3]