**Sort Array ascending without using sort function**

```
function bubbleSort(array) {
  var done = false;
  while (!done) {
    done = true;
    for (var i = 1; i < array.length; i += 1) {
      if (array[i - 1] > array[i]) {
        done = false;
        var tmp = array[i - 1];
        array[i - 1] = array[i];
        array[i] = tmp;
      }
    }
  }
  return array;
}
var numbers = [12, 10, 15, 11, 14, 13, 16];
bubbleSort(numbers);
console.log(numbers);
```

**********

```
const fruits = [11, 0, 4, 2, 1];
fruits.sort((a, b) => {
  return a-b;
});
console.log(fruits);
```

**Sort Array decending without using sort function**

```
function bubbleSort(array) {
  var done = false;
  while (!done) {
    done = true;
    for (var i = 1; i < array.length; i += 1) {
      if (array[i - 1] > array[i]) {
        done = false;
        var tmp = array[i - 1];
        array[i - 1] = array[i];
        array[i] = tmp;
```

```
            }
           }
          }
          return array;
        }
        var numbers = [12, 10, 15, 11, 14, 13, 16];
        bubbleSort(numbers);
        console.log(numbers);
****************************************
        const fruits = [11, 0, 4, 2, 1];
        fruits.sort((a, b) => {
          return b - a;
        });
        console.log(fruits);
```

**Find Vowels in a string**

```
        function findVowel(str) {
          var vowels = "aeiou";
          var vowelout = [];
          for (var i = 0; i < str.length; i++) {
            if (vowels.indexOf(str[i].toLowerCase()) != -1) {
              vowelout.push(str[i]);
            }
          }
          return vowelout;
        }
        console.log(findVowel("sdfhsAEIOUjbvcxvdfaambfghsarg"));
```

**Remove Duplicates in Array**

```
        var numbers = [12, 10, 15, 11, 14, 13, 16, 12, 10, 11, 11];
        var uniqueNum = [];
        for (var i = 0; i < numbers.length; i++) {
          if (uniqueNum.indexOf(numbers[i]) === -1) {
            uniqueNum.push(numbers[i]);
          }
        }
        console.log(uniqueNum);
******************************
```

```
const names = ['John', 'Paul', 'George', 'Ringo', 'John'];
let unique = [...new Set(names)];
console.log(unique);
```

**********

```
let arr = [1,2,3,1,1,1,4,5];
let filtered = arr.filter((item,index) => arr.indexOf(item) === index);
 console.log(filtered)
```

**Remove Elements in Array**

```
const array = [2, 5, 9];
const index = array.indexOf(15);
if (index > -1) {
  array.splice(index, 1); // 2nd parameter means remove one item only
}
console.log(array);
```

**Find index of element in array**

```
const array = [2, 5, 9];
const index = array.indexOf(15);
console.log(index);
```

**Reverse a string without in build methods**

```
var str = "shiva kumar";
var newStr = '';
for (let i=str.length-1;i>=0;i--){
  newStr += str[i];
}
console.log(newStr);
```

**********

```
var str = "shiva kumar";
var newStr = str.split('').reverse().join('');
console.log(newStr);
```

**Find max/min number in an array**

```
var numbers = [12, 10, 15, 11, 14, 13, 16, 12, 10, 11, 11];
numbers.sort((a,b) => {
  return a-b;
```

```
})
console.log(numbers[0]); // min
console.log(numbers[numbers.length-1]); // max
```

**Check whether 2 strings are made up of same characters and length**

```
function check(a,b){
 if(a.split('').sort().join('') === b.split('').sort().join('')){
  return 'same';
 } else {
  return 'not same';
 }
}
console.log(check('sir','ris'));
```

**Check whether 2 strings are made up of same characters**

```
function check(a,b){
 if([...new Set(a.split(''))].sort().join('') === [...new
Set(b.split(''))].sort().join('')){
  return 'same';
 } else {
  return 'not same';
 }
}
console.log(check('sir','riiiidgdfgs'));
```

**Sort Array Regardless of Negative/positive number**

```
const numbers = [11, 0, 4, 2, 1, -2, -11, -3];
numbers.sort((a, b) => {
 return Math.abs(a) - Math.abs(b);
});
console.log(numbers);
```

**How to get a Key names and value from an array of objects**

```
var arr = [{name:'shiva',id:12},
      {name:'kumar',id:22},
      {name:'tm',id:32}];
var key = '';
```

```
var val = '';
arr.forEach((ar) =>{
  key += Object.keys(ar);
  val += Object.values(ar);
})
console.log(key);
console.log(val);
```

**Find the Longest word in the Sentence**

```
var str ="hi my name is shiva kumar TM";
str = str.split(' ').sort((a,b)=>{
  return b.length - a.length;
});
console.log(str[0]);
console.log(str[str.length-1]);
```

**Check whether a string is palindrome or Not**

```
function palindrom(str) {
  if (str.split('').reverse().join('') === str) {
    return true;
  } else {
    return false;
  }
}
console.log(palindrom("level"));
```

**Find the words which has “i”**

```
var arr = ["shiva", "kumar", "sunitha", "aadya"];
var result = [];
arr.forEach((ar) => {
  if (ar.includes("i")) {
    result.push(ar);
  }
});
console.log(result);
```

**Remove last character from string**

```
let str = "Hello Worlds";
str = str.slice(0, -1);
console.log(str);
```

**Factorial of n**

```
const recursiceFactorial = (num, res = 1) => {
 if (num) {
  return recursiceFactorial(num - 1, res * num);
 }
 return res;
};
console.log(recursiceFactorial(3));
```

**Write a function that performs binary search on a sorted array.**

```
function binarySearch(arr,value,startPos,endPos){
 if(startPos > endPos) return -1;
 let middleIndex = Math.floor(startPos+endPos)/2;
 if(arr[middleIndex] === value) return middleIndex;
 elsif(arr[middleIndex > value]){
  return binarySearch(arr,value,startPos,middleIndex-1);
 }
 else{
  return binarySearch(arr,value,middleIndex+1,endPos);
 }
}
```

**Implement a function that returns an updated array with r right rotations on an array of integers a .**
**Example: Given the following array: [2,3,4,5,7]**
**Perform 3 right rotations: First rotation : [7,2,3,4,5] , Second rotation : [5,7,2,3,4] and, Third rotation: [4,5,7,2,3] return [4,5,7,2,3]**

```
function rotateRight(arr,rotations){
 if(rotations == 0) return arr;
 for(let i = 0; i < rotations;i++){
  let element = arr.pop();
  arr.unshift(element);
 }
```

```
  return arr;
}
rotateRight([2, 3, 4, 5, 7], 3); // Return [4,5,7,2,3]
rotateRight([44, 1, 22, 111], 5); // Returns [111,44,1,22]
```