

## 1)Print odd numbers in an array

```
anonymous : function(array){  
    for(var i = 0 ; i< array.length ; i++){  
        if(array[i]%2!=0){  
            console.log(array[i])  
        }  
    }  
}
```

```
IIFE :    (function(array){  
    for(var i = 0 ; i< array.length ; i++){  
        if(array[i]%2!=0){  
            console.log(array[i])  
        }  
    }  
})([1,2,3,4])
```

```
Arrow Function  oddNumbers = (array) => {  
    for(var i = 0 ; i< array.length ; i++){  
        if(array[i]%2!=0){  
            console.log(array[i])  
        }  
    }  
}
```

## 2)Convert all the strings to title caps in a string array

```
anonymous : function (str) {  
    str = str.toLowerCase().split(' ');  
    for (var i = 0; i < str.length; i++) {  
        str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1);  
    }  
}
```

```

        return str.join(' ');
    }

    IIFE : (function (str) {

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

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

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

        }

        return str.join(' ');

    })("MUDRA IS MY NAME");

```

```

    Arrow Function : titleCase = (str) => {

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

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

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

        }

        return str.join(' ');

    }

```

### 3)Sum of all numbers in an array

```

    anonymous : function(array){

        var sum = 0;

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

            sum = sum + array[i];

        }

        return sum;

    }

```

```

    IIFE : (function(array){

        var sum = 0;

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

            sum = sum + array[i];

        }

        return sum;

    })([1,2,3,4])

```

Arrow:     sum = (array)=>{

```

    var sum = 0;

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

        sum = sum + array[i];

    }

    return sum;

}
```

#### 4)Return all the prime numbers in an array

Anonymous Function:

```

function(numArray){

    numArray = numArray.filter((number) => {

        for (var i = 2; i <= Math.sqrt(number); i++) {

            if (number % i === 0) return false;

        }

        return true;

    });

    console.log(numArray);

}
```

IIFE

```

(

function(numArray){

    numArray = numArray.filter((number) => {

        for (var i = 2; i <= Math.sqrt(number); i++) {

            if (number % i === 0) return false;

        }

        return true;

    });

    console.log(numArray);

})([1,2,3,4])
```

Arrow Function :

```

primeNumber = (numArray) => {
    numArray = numArray.filter((number) => {
        for (var i = 2; i <= Math.sqrt(number); i++) {
            if (number % i === 0) return false;
        }
        return true;
    });
    console.log(numArray);
}

```

## 5) Return all the palindromes in an array

```

function isPalindrome(N)
{
    let str = "" + N;
    let len = str.length;
    for (let i = 0; i < parseInt(len / 2, 10); i++)
    {
        if (str[i] != str[len - 1 - i])
            return false;
    }
    return true;
}

```

Anonymous Function : function (arr, n)

```

{
    // Traversing each element of the array
    // and check if it is palindrome or not
    for (let i = 0; i < n; i++)
    {
        let ans = isPalindrome(arr[i]);
    }
}

```

```

        if (ans == false)
            return false;
    }
    return true;
}

```

IIFE :

```

    ( function (arr, n)
    {
        // Traversing each element of the array
        // and check if it is palindrome or not
        for (let i = 0; i < n; i++)
        {
            let ans = isPalindrome(arr[i]);
            if (ans == false)
                return false;
        }
        return true;
    })([1,2,3] , 3)

```

Arrow :

Palindrome = (arr, n) =>

```

{
    // Traversing each element of the array
    // and check if it is palindrome or not
    for (let i = 0; i < n; i++)
    {
        let ans = isPalindrome(arr[i]);
        if (ans == false)
            return false;
    }
}

```

```
    }  
    return true;  
}
```

## 6.Return median of two sorted arrays of the same size.

Anonymous function :

```
var getMedian =function (ar1, ar2, n)
```

```
{  
    var i = 0;  
    var j = 0;  
    var count;  
    var m1 = -1, m2 = -1;  
    for (count = 0; count <= n; count++)  
    {  
        if (i == n)  
        {  
            m1 = m2;  
            m2 = ar2[0];  
            break;  
        }  
        else if (j == n)  
        {  
            m1 = m2;  
            m2 = ar1[0];  
            break;  
        }  
        if (ar1[i] <= ar2[j])  
        {  
            m1 = m2;  
            m2 = ar1[i];  
            i++;  
        }  
    }  
}
```

```

    }
    else
    {
        m1 = m2;
        m2 = ar2[j];
        j++;
    }
}

return (m1 + m2)/2;
}

var ar1 = [1, 12, 15, 26, 38];
var ar2 = [2, 13, 17, 30, 45];
var n1 = ar1.length;
var n2 = ar2.length;
if (n1 == n2)
    document.write("Median is "+ getMedian(ar1, ar2, n1));
else
    document.write("Doesn't work for arrays of unequal size");
IIFE :
var getMedian = (ar1, ar2, n)=>
{
    var i = 0;
    var j = 0;
    var count;
    var m1 = -1, m2 = -1;
    for (count = 0; count <= n; count++)
    {
        if (i == n)
        {
            m1 = m2;

```

```

        m2 = ar2[0];
        break;
    }
    else if (j == n)
    {
        m1 = m2;
        m2 = ar1[0];
        break;
    }
    if (ar1[i] <= ar2[j])
    {
        m1 = m2;
        m2 = ar1[i];
        i++;
    }
    else
    {
        m1 = m2;
        m2 = ar2[j];
        j++;
    }
}

return (m1 + m2)/2;
}

var ar1 = [1, 12, 15, 26, 38];
var ar2 = [2, 13, 17, 30, 45];
var n1 = ar1.length;
var n2 = ar2.length;
if (n1 == n2)
    document.write("Median is " + getMedian(ar1, ar2, n1));

```



else

```
document.write("Doesn't work for arrays of unequal size");
```

## 7. Remove duplicates from an Array

```
Anonymous Function : function(array){  
    let dup = [...new Set(array)];  
    console.log(dup);  
}
```

```
IIFE :      (function(array){  
    let dup = [...new Set(array)];  
    console.log(dup);  
})([1,1,2,3,4])
```

## 8. Rotate an array by K times

```
function reverse(array , li , ri){  
    while(li < ri){  
        int temp = a[li];  
        a[li]= a[ri];  
        a[ri] = temp;  
  
        li++;  
        ri--;  
    }  
}
```

```
Anonymous function : function(array , k){  
    k = k % a.length;  
    if(k < 0){  
        k += a.length;  
    }  
}
```

```
reverse(a, 0, a.length - k - 1);  
reverse(a, a.length - k, a.length - 1);  
reverse(a, 0, a.length - 1);  
}
```

```
IIFE : (function(array , k){  
    k = k % a.length;  
    if(k < 0){  
        k += a.length;  
    }  
  
    reverse(a, 0, a.length - k - 1);  
    reverse(a, a.length - k, a.length - 1);  
    reverse(a, 0, a.length - 1);  
})([1,2,3,4] , 2)
```