Print odd numbers in a array

Anonymous Function

var odd=function(a)

{

    for(var i=0;i<a.length;i++)

    {

        if(a[i]%2!==0)

        {

            console.log("Odd Numbers",a[i]);

        }

    }

}

odd([1,2,3,4]);

IIFE

(function odd(a)

{

    for(var i=0;i<a.length;i++)

    {

        if(a[i]%2!==0)

        {

            console.log("Odd Numbers",a[i]);

        }

    }

})

odd([1,2,3,4,5]);

Arrow Function

var odd=(a)=>{

    for(var i=0;i<a.length;i++)

    {

        if(a[i]%2!==0)

        {

            console.log("Odd Numbers",a[i]);

        }

    }

}

odd([1,2,3,4,5]);

Convert all the strings to title caps in a string array

Anonymous Function

var caps= function(s)

{

    s = s.toLowerCase().split(' ');

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

    {

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

    }

    console.log(s.join(' '));

}

caps("im a software developer");

IIFE

(function (s)

{

    s = s.toLowerCase().split(' ');

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

    {

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

    }

    console.log(s.join(' '));

})

("im full stack developer");

Arrow Function

var cap=(s)=>{

    s = s.toLowerCase().split(' ');

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

    {

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

    }

    console.log(s.join(' '));

}

cap("im going to study hard");

Sum of all numbers in an array

Anonymous Function

var sum=function(z)

{

    var x=0;

    for(var i=0;i<z.length;i++)

    {

        x+=z[i];

    }

    console.log(x);

}

sum([1,2,3,4,5]);

IIFE

(function(s)

{

    var x=0;

    for(var i=0;i<s.length;i++)

    {

        x+=s[i];

    }

    console.log(x);

})

([1,2,3,4,5]);

Arrow Function

var sum=(z)=>

{

    var x=0;

    for(var i=0;i<z.length;i++)

    {

        x+=z[i];

    }

    console.log(x);

}

sum([1,2,3,4,5]);

Return all the prime numbers in an array

Anonymous Function

var a = [1,2,3,4,5,6,7]

var prime = [];

var store=function(num) {

      for (var j = 2; j <= num/2; j++)

       {

       if ((num % j) == 0)

        {

        return false;

       }

     }

     return true;

}

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

  if (store(a[i])) {

      prime.push(a[i])

  }

}

console.log(prime);

IIFE

var a = [1,2,3,4,5,6,7]

var prime = [];

(function(num) {

      for (var j = 2; j <= num/2; j++)

       {

       if ((num % j) == 0){

        return false;

       }

     }

     return true;

})

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

{

  if (store(a[i]))

  {

      prime.push(a[i])

  }

}

console.log(prime);

Arrow Function

var a = [1,2,3,4,5,6,7]

var prime = [];

var we=(num)=> {

      for (var j = 2; j <= num/2; j++)

       {

       if ((num % j) == 0)

        {

        return false;

       }

     }

     return true;

}

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

{

  if (store(a[i]))

  {

      prime.push(a[i])

  }

}

console.log(prime);

Return all the palindromes in an array

Anonymous Function

const wordsArray = ["level", "hello", "deed", "world", "civic", "madam"];

const palindrome = function(arr) {

    return arr.filter(function(word) {

        return word === word.split("").reverse().join("");

    });

}(wordsArray);

console.log(palindrome);

IIFE

const words= ["level", "hello", "deed", "world", "civic", "madam"];

const palindromes = (function(arr) {

    return arr.filter(function(word) {

        return word === word.split("").reverse().join("");

    });

})(words);

console.log(palindromes);

Return median of two sorted arrays of the same size

Anonymous Function

const Median = function() {

    return function(arr1, arr2) {

      const mergedArray = arr1.concat(arr2).sort((a, b) => a - b);

      const length = mergedArray.length;

      if (length % 2 === 0) {

        const midIndex1 = length / 2 - 1;

        const midIndex2 = length / 2;

        return (mergedArray[midIndex1] + mergedArray[midIndex2]) / 2;

      } else {

        const midIndex = Math.floor(length / 2);

        return mergedArray[midIndex];

      }

    };

  }();

  const ar1 = [1, 3, 5];

  const ar2 = [2, 4, 6];

  const medians = Median(ar1, ar2);

  console.log(medians);

IIFE

const findMedian = (function() {

    return function(arr1, arr2) {

      const mergedArray = arr1.concat(arr2).sort((a, b) => a - b);

      const length = mergedArray.length;

      if (length % 2 === 0) {

        const midIndex1 = length / 2 - 1;

        const midIndex2 = length / 2;

        return (mergedArray[midIndex1] + mergedArray[midIndex2]) / 2;

      } else {

        const midIndex = Math.floor(length / 2);

        return mergedArray[midIndex];

      }

    };

  })();

  const array1 = [1, 3, 5];

  const array2 = [2, 4, 6];

  const median = findMedian(array1, array2);

  console.log(median);

Remove duplicates in an array

Anonymous Function

let arr = ["hello","hai","welcome","hello","hai","going","welcome"];

function duplicate(arr) {

    return [...new Set(arr)];

}

console.log(duplicate(arr));

IIFE

(function(array){

    let dup = [...new Set(array)];

   })(["hello","hai","welcome","hello","hai","going","welcome"])

Rotate array by k times

Anonymous Function

const rotateArrays = function () {

    return function (arr, k) {

      const len = arr.length;

      k = k % len; // Ensure k is within the array length

      return arr.slice(-k).concat(arr.slice(0, len - k));

    };

  }();

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

  const rotatedArrays = rotateArrays(originalArrays, 2);

  console.log(rotatedArrays)

IIFE

const rotateArray = (function () {

    return function (arr, k) {

      const len = arr.length;

      k = k % len; // Ensure k is within the array length

      return arr.slice(-k).concat(arr.slice(0, len - k));

    };

  })();

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

  const rotatedArray = rotateArray(originalArray, 2);