|  |  |
| --- | --- |
|  | **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]) |
|  | } |
|  | } |
|  | } |
|  |  |
|  | **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(' '); |
|  | } |
|  | **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; |
|  | } |
|  | **Return all the prime numbers in an array** |
|  | **ANONYMOUS**: |
|  | 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**: |
|  |  |
|  | 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); |
|  | } |
|  |  |
|  | **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 (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; |
|  | }  **Median of two sorted array in same size:**  const median=function(a,b)  {  let c=[…a,…b].sort(function(a,b){return(a-b)});  Const half=c.length/2|0;  If(c.length%2)return c[half];  Return c[half] +c[half-1]/2;  }  Const arr=[1.2,44,44,12]  Const arr2=[5,6,7,87,12]  Console.log(median(arr,arr2)); |
|  | **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]) |
|  |  |
|  |  |
|  | **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(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) |