1. Print odd numbers is an array in anonymous, IIFE, arrow:

|  |
| --- |
| 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(' '); |
| })("IMRAN 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); |
| 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 of palindromes:   |  |  | | --- | --- | |  | { | |  | 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) | |  | for (let i = 0; i < n; i++) | |  | { | |  | let ans = isPalindrome(arr[i]); | |  | if (ans == false) | |  | return false; | |  | } | |  | return true; | |  | IIFE  ( function (arr, n) | |  | for (let i = 0; i < n; i++) | |  | { | |  | let ans = isPalindrome(arr[i]); | |  | if (ans == false) | |  | return false; | |  | } | |  | return true; | |  | Arrow : | |  | Palindrome = (arr, n) => | |  | {  for (let i = 0; i < n; i++) | |  | { | |  | let ans = isPalindrome(arr[i]); | |  | if (ans == false) | |  | return false; | |  | } | |  | return true;  } | |  |  | |  |  | |  | |  | |  | | |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | } |
|  |
|  |
|  |
|  |
|  |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |