1. Print odd numbers in an array:

**Anonymous Function:**

let arrayNumber=[1,2,3,4,5,6,7,8,9,0];

function OddNumber(array){

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

if(array[i]%2!=0){

console.log(array[i])

}

}

}

OddNumber(arrayNumber)

**IIFE Function:**

(function(array){

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

if(array[i]%2!=0){

console.log(array[i])

}

}

})([1,2,3,4,5,6,7,8,9,10,11])

**Arrow Function:**

let arrayfunction = (array) => {

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

if(array[i]%2!=0){

console.log(array[i])

}

}

}

arrayfunction([1,2,3,4,5,6,7,8,9,0])

1. Convert all the strings to title caps in a string array:

**Anonymous Function :**

function titleCaps(string) {

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

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

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

}

return string.join(' ');

}

console.log("Title Caps using anonymous",titleCaps("hello world!"))

**IIFE :**

let resultTitle = (function titleCapsIIFE(string) {

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

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

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

}

return string.join(' ');

})("HELLO WORLD");

console.log(resultTitle);

**Arrow Function :**

let titleCapsArrow = (string) => {

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

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

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

}

return string.join(' ');

}

console.log("Title Caps using arrow",titleCapsArrow("hello world!"))

1. Sum of all numbers in an array

**Anonymous Function:**

function sumNumber(array){

var sum = 0;

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

sum = sum + array[i];

}

return sum;

}

console.log(sumNumber([21,34,45,34,1,45,3]))

**IIFE :**

let sum = (function(array){

var sum = 0;

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

sum = sum + array[i];

}

return sum;

})([21,34,45,34,1,45,3])

console.log(sum)

**Arrow Function:**

let sumNumberArrow = (array)=>{

var sum = 0;

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

sum = sum + array[i];

}

return sum;

}

console.log(sumNumberArrow([21,34,45,34,1,45,3]))

1. Return all the prime numbers in an array

Anonymous function:

const numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

const primeNumbersAnonymous = function(arr) {

return arr.filter(function(num) {

for (let i = 2; i <= num / 2; i++) {

if (num % i === 0) {

return false;

}

}

return num > 1;

});

};

console.log(primeNumbersAnonymous(numbers)); // Output: [2, 3, 5, 7, 11, 13]

IIFE function :

const numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

const primeNumbersIIFE = (function() {

return function(arr) {

return arr.filter(function(num) {

for (let i = 2; i <= num / 2; i++) {

if (num % i === 0) {

return false;

}

}

return num > 1;

});

};

})();

console.log(primeNumbersIIFE(numbers)); // Output: [2, 3, 5, 7, 11, 13]

Arrow Function

const numbers = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];

const primeNumbersArrow = (() => {

return arr => arr.filter(num => {

for (let i = 2; i <= num / 2; i++) {

if (num % i === 0) {

return false;

}

}

return num > 1;

});

})();

console.log(primeNumbersArrow(numbers)); // Output: [2, 3, 5, 7, 11, 13]

1. Return all the palindromes in an array

Anonymous Function :

const arr = ["madam", "hello", "level", "world", "racecar", "noon"];

const palindromesAnonymous = function(arr) {

return arr.filter(function(word) {

const reversedWord = word.split('').reverse().join('');

return word === reversedWord;

});

};

console.log(palindromesAnonymous(arr)); // Output: ["madam", "level", "racecar", "noon"]

IIFE Function:

const arr = ["madam", "hello", "level", "world", "racecar", "noon"];

const palindromesIIFE = (function() {

return function(arr) {

return arr.filter(function(word) {

const reversedWord = word.split('').reverse().join('');

return word === reversedWord;

});

};

})();

console.log(palindromesIIFE(arr)); // Output: ["madam", "level", "racecar", "noon"]

Arrow Function:

const arr = ["madam", "hello", "level", "world", "racecar", "noon"];

const palindromesArrow = (() => {

return arr => arr.filter(word => {

const reversedWord = word.split('').reverse().join('');

return word === reversedWord;

});

})();

console.log(palindromesArrow(arr)); // Output: ["madam", "level", "racecar", "noon"]

1. Return median of two sorted arrays of the same size.

const arr1 = [1, 3, 5];

const arr2 = [2, 4, 6];

// Anonymous function

const medianAnonymous = function(arr1, arr2) {

const mergedArr = arr1.concat(arr2);

const sortedArr = mergedArr.sort((a, b) => a - b);

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

if (sortedArr.length % 2 === 0) {

return (sortedArr[midIndex - 1] + sortedArr[midIndex]) / 2;

} else {

return sortedArr[midIndex];

}

};

console.log(medianAnonymous(arr1, arr2)); // Output: 3.5

// IIFE (Immediately Invoked Function Expression)

const medianIIFE = (function() {

return function(arr1, arr2) {

const mergedArr = arr1.concat(arr2);

const sortedArr = mergedArr.sort((a, b) => a - b);

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

if (sortedArr.length % 2 === 0) {

return (sortedArr[midIndex - 1] + sortedArr[midIndex]) / 2;

} else {

return sortedArr[midIndex];

}

};

})();

console.log(medianIIFE(arr1, arr2)); // Output: 3.5

// Arrow function

const medianArrow = (() => {

return (arr1, arr2) => {

const mergedArr = arr1.concat(arr2);

const sortedArr = mergedArr.sort((a, b) => a - b);

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

if (sortedArr.length % 2 === 0) {

return (sortedArr[midIndex - 1] + sortedArr[midIndex]) / 2;

} else {

return sortedArr[midIndex];

}

};

})();

console.log(medianArrow(arr1, arr2)); // Output: 3.5

1. Return median of two sorted arrays of the same size.

const arr = [1, 2, 2, 3, 4, 4, 5, 5, 6];

// Anonymous function

const removeDuplicatesAnonymous = function(arr) {

return arr.filter(function(item, index, self) {

return self.indexOf(item) === index;

});

};

console.log(removeDuplicatesAnonymous(arr)); // Output: [1, 2, 3, 4, 5, 6]

// IIFE (Immediately Invoked Function Expression)

const removeDuplicatesIIFE = (function() {

return function(arr) {

return arr.filter(function(item, index, self) {

return self.indexOf(item) === index;

});

};

})();

console.log(removeDuplicatesIIFE(arr)); // Output: [1, 2, 3, 4, 5, 6]

// Arrow function

const removeDuplicatesArrow = (() => {

return arr => arr.filter((item, index, self) => self.indexOf(item) === index);

})();

console.log(removeDuplicatesArrow(arr)); // Output: [1, 2, 3, 4, 5, 6]

1. Rotate an array by k times

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

const k = 2; // Number of rotations

// Anonymous function

const rotateArrayAnonymous = function(arr, k) {

const rotatedArray = arr.slice(k).concat(arr.slice(0, k));

return rotatedArray;

};

console.log(rotateArrayAnonymous(arr, k)); // Output: [3, 4, 5, 1, 2]

// IIFE (Immediately Invoked Function Expression)

const rotateArrayIIFE = (function() {

return function(arr, k) {

const rotatedArray = arr.slice(k).concat(arr.slice(0, k));

return rotatedArray;

};

})();

console.log(rotateArrayIIFE(arr, k)); // Output: [3, 4, 5, 1, 2]

// Arrow function

const rotateArrayArrow = (() => {

return (arr, k) => {

const rotatedArray = arr.slice(k).concat(arr.slice(0, k));

return rotatedArray;

};

})();

console.log(rotateArrayArrow(arr, k)); // Output: [3, 4, 5, 1, 2]