1. Programs in anonymous function & IIFE
2. Print odd numbers in an array

// Using an anonymous function

const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9];

const printOddNumbers = function(arr) {

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

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

console.log(arr[i]);

}

}

};

printOddNumbers(arr);

// Using an IIFE (Immediately Invoked Function Expression)

const arr2 = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19];

(function(arr) {

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

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

console.log(arr[i]);

}

}

})(arr2);

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

// Using an anonymous function

const stringArray = ["hello world", "javascript is fun", "title caps"];

const convertToTitleCaps = function(arr) {

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

arr[i] = arr[i].toLowerCase().split(' ').map(word => word.charAt(0).toUpperCase() + word.slice(1)).join(' ');

}

};

convertToTitleCaps(stringArray);

console.log(stringArray);

// Using an IIFE (Immediately Invoked Function Expression)

const stringArray2 = ["another example", "more strings", "in title case"];

(function(arr) {

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

arr[i] = arr[i].toLowerCase().split(' ').map(word => word.charAt(0).toUpperCase() + word.slice(1)).join(' ');

}

})(stringArray2);

console.log(stringArray2);

1. Sum of all numbers in an array

// Using an anonymous function

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

const calculateSum = function(arr) {

let sum = 0;

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

sum += arr[i];

}

return sum;

};

const sum1 = calculateSum(numArray);

console.log("Sum (using anonymous function):", sum1);

// Using an IIFE (Immediately Invoked Function Expression)

const numArray2 = [10, 20, 30, 40, 50];

const sum2 = (function(arr) {

let sum = 0;

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

sum += arr[i];

}

return sum;

})(numArray2);

console.log("Sum (using IIFE):", sum2);

d.Return all the prime numbers in an array

// Using an anonymous function

const numArray = [1, 2, 3, 4, 5, 6, 7, 8, 9];

const findPrimeNumbers = function(arr) {

const isPrime = (num) => {

if (num <= 1) return false;

if (num <= 3) return true;

if (num % 2 === 0 || num % 3 === 0) return false;

for (let i = 5; i \* i <= num; i += 6) {

if (num % i === 0 || num % (i + 2) === 0) return false;

}

return true;

};

return arr.filter((num) => isPrime(num));

};

const primeNumbers = findPrimeNumbers(numArray);

console.log("Prime numbers (using anonymous function):", primeNumbers);

// Using an IIFE (Immediately Invoked Function Expression)

const numArray2 = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19];

const primeNumbers2 = (function(arr) {

const isPrime = (num) => {

if (num <= 1) return false;

if (num <= 3) return true;

if (num % 2 === 0 || num % 3 === 0) return false;

for (let i = 5; i \* i <= num; i += 6) {

if (num % i === 0 || num % (i + 2) === 0) return false;

}

return true;

};

return arr.filter((num) => isPrime(num));

})(numArray2);

console.log("Prime numbers (using IIFE):", primeNumbers2);

e.Return all the palindromes in an array

// Using an anonymous function

const stringArray = ["level", "hello", "racecar", "world", "deified"];

const findPalindromes = function(arr) {

const isPalindrome = (str) => {

const reversed = str.split("").reverse().join("");

return str === reversed;

};

return arr.filter((str) => isPalindrome(str));

};

const palindromes = findPalindromes(stringArray);

console.log("Palindromes (using anonymous function):", palindromes);

// Using an IIFE (Immediately Invoked Function Expression)

const stringArray2 = ["madam", "apple", "noon", "banana", "civic"];

const palindromes2 = (function(arr) {

const isPalindrome = (str) => {

const reversed = str.split("").reverse().join("");

return str === reversed;

};

return arr.filter((str) => isPalindrome(str));

})(stringArray2);

console.log("Palindromes (using IIFE):", palindromes2);

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

// Using an anonymous function

const array1 = [1, 3, 5];

const array2 = [2, 4, 6];

const findMedian = function(arr1, arr2) {

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

const middleIndex = Math.floor(mergedArray.length / 2);

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

// If the merged array has an even length, return the average of the middle two elements.

return (mergedArray[middleIndex - 1] + mergedArray[middleIndex]) / 2;

} else {

// If the merged array has an odd length, return the middle element.

return mergedArray[middleIndex];

}

};

const median = findMedian(array1, array2);

console.log("Median (using anonymous function):", median);

// Using an IIFE (Immediately Invoked Function Expression)

const array3 = [10, 12, 14];

const array4 = [11, 13, 15];

const median2

g. Remove duplicates from an array.

// Using an anonymous function

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

const removeDuplicates = function(arr) {

return arr.filter((value, index, self) => {

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

});

};

const uniqueArray = removeDuplicates(originalArray);

console.log("Array with duplicates removed (using anonymous function):", uniqueArray);

// Using an IIFE (Immediately Invoked Function Expression)

const originalArray2 = [10, 20, 20, 30, 40, 40, 50, 50, 60];

const uniqueArray2 = (function(arr) {

return arr.filter((value, index, self) => {

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

});

})(originalArray2);

console.log("Array with duplicates removed (using IIFE):", uniqueArray2);

h. Rotate an array by k times.

// Using an anonymous function

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

const k = 2;

const rotateArray = function(arr, k) {

const n = arr.length;

const rotatedArray = [];

for (let i = 0; i < n; i++) {

const newIndex = (i + k) % n;

rotatedArray[newIndex] = arr[i];

}

return rotatedArray;

};

const rotated = rotateArray(array, k);

console.log("Rotated array (using anonymous function):", rotated);

// Using an IIFE (Immediately Invoked Function Expression)

const array2 = [10, 20, 30, 40, 50];

const k2 = 3;

const rotated2 = (function(arr, k) {

const n = arr.length;

const rotatedArray = [];

for (let i = 0; i < n; i++) {

const newIndex = (i + k) % n;

rotatedArray[newIndex] = arr[i];

}

return rotatedArray;

})(array2, k2);

console.log("Rotated array (using IIFE):", rotated2);

1. Programs in arrow functions.
2. Print odd numbers in an array

const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9];

const printOddNumbers = (arr) => {

arr.forEach((num) => {

if (num % 2 !== 0) {

console.log(num);

}

});

};

printOddNumbers(arr);

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

const stringArray = ["hello world", "javascript is fun", "title caps"];

const convertToTitleCaps = (arr) => {

return arr.map((str) => {

return str

.toLowerCase()

.split(' ')

.map((word) => word.charAt(0).toUpperCase() + word.slice(1))

.join(' ');

});

};

const titleCapsArray = convertToTitleCaps(stringArray);

console.log(titleCapsArray);

1. Sum of all numbers in an array.

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

const calculateSum = (arr) => {

return arr.reduce((accumulator, currentValue) => accumulator + currentValue, 0);

};

const sum = calculateSum(numArray);

console.log("Sum of numbers:", sum);

1. Return all the prime numbers in an array.

const numArray = [1, 2, 3, 4, 5, 6, 7, 8, 9];

const isPrime = (num) => {

if (num <= 1) return false;

if (num <= 3) return true;

if (num % 2 === 0 || num % 3 === 0) return false;

for (let i = 5; i \* i <= num; i += 6) {

if (num % i === 0 || num % (i + 2) === 0) return false;

}

return true;

};

const findPrimeNumbers = (arr) => {

return arr.filter((num) => isPrime(num));

};

const primeNumbers = findPrimeNumbers(numArray);

console.log("Prime numbers:", primeNumbers);

1. Return all the palindromes in an array.

const stringArray = ["level", "hello", "racecar", "world", "deified"];

const isPalindrome = (str) => {

const reversed = str.split("").reverse().join("");

return str === reversed;

};

const findPalindromes = (arr) => {

return arr.filter((str) => isPalindrome(str));

};

const palindromes = findPalindromes(stringArray);

console.log("Palindromes:", palindromes);