1. **Do the below 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, 10];

const printOddNumbers = function(arr) {

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

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

console.log(arr[i]);

}

}

};

printOddNumbers(arr);

* **Using IIFE:**

const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

(function(arr) {

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

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

console.log(arr[i]);

}

}

})(arr);

* **Both versions will output the odd numbers from the array [1, 3, 5, 7, 9].**

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

* **Using an anonymous function:**

const strings = ["hello", "world", "javascript", "programming"];

const convertToTitleCaps = function(strings) {

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

strings[i] = strings[i][0].toUpperCase() + strings[i].substring(1).toLowerCase();

}

return strings;

};

console.log(convertToTitleCaps(strings));

* **Using IIFE:**

const strings = ["hello", "world", "javascript", "programming"];

(function(strings) {

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

strings[i] = strings[i][0].toUpperCase() + strings[i].substring(1).toLowerCase();

}

console.log(strings);

})(strings);

* **Both versions will convert the strings to title caps, resulting in ["Hello", "World", "Javascript", "Programming"].**

1. **Sum of all numbers in an array**

* **Using an anonymous function:**

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

const sumOfNumbers = function(numbers) {

let sum = 0;

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

sum += numbers[i];

}

return sum;

};

console.log(sumOfNumbers(numbers));

* **Using IIFE:**

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

(function(numbers) {

let sum = 0;

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

sum += numbers[i];

}

console.log(sum);

})(numbers);

* **Both versions will output the sum of all numbers in the array, which is 15 for the provided array [1, 2, 3, 4, 5].**

1. **Return all the prime numbers in an array**

* **Using an anonymous function:**

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const isPrime = function(num) {

if (num <= 1) {

return false;

}

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) {

return false;

}

}

return true;

};

const primeNumbers = function(numbers) {

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

};

console.log(primeNumbers(numbers));

* **Using IIFE:**

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

(function(numbers) {

const isPrime = function(num) {

if (num <= 1) {

return false;

}

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) {

return false;

}

}

return true;

};

const primeNumbers = numbers.filter(num => isPrime(num));

console.log(primeNumbers);

})(numbers);

* **Both versions will output an array containing all the prime numbers from the provided array [2, 3, 5, 7].**

1. **Return all the palindromes in an array**

* Using an anonymous function:

const words = ["hello", "level", "world", "radar", "javascript"];

const isPalindrome = function(word) {

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

return word === reversedWord;

};

const palindromeWords = function(words) {

return words.filter(word => isPalindrome(word));

};

console.log(palindromeWords(words));

* **Using IIFE:**

const words = ["hello", "level", "world", "radar", "javascript"];

(function(words) {

const isPalindrome = function(word) {

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

return word === reversedWord;

};

const palindromeWords = words.filter(word => isPalindrome(word));

console.log(palindromeWords);

})(words);

* **Both versions will output an array containing all the palindrome words from the provided array ["level", "radar"].**

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

* **Using an anonymous function:**

const findMedianSortedArrays = function(nums1, nums2) {

const merged = nums1.concat(nums2).sort((a, b) => a - b);

const length = merged.length;

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

if (length % 2 === 0) {

return (merged[mid - 1] + merged[mid]) / 2;

} else {

return merged[mid];

}

};

const nums1 = [1, 3, 5];

const nums2 = [2, 4, 6];

console.log(findMedianSortedArrays(nums1, nums2));

* **Using IIFE:**

const nums1 = [1, 3, 5];

const nums2 = [2, 4, 6];

(function(nums1, nums2) {

const merged = nums1.concat(nums2).sort((a, b) => a - b);

const length = merged.length;

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

if (length % 2 === 0) {

console.log((merged[mid - 1] + merged[mid]) / 2);

} else {

console.log(merged[mid]);

}

})(nums1, nums2);

* **Both versions will output the median of the merged array, which is 3.5 for the provided arrays [1, 3, 5] and [2, 4, 6].**

1. **Remove duplicates from an array**

* **Using an anonymous function:**

const removeDuplicates = function(arr) {

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

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

});

};

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

console.log(removeDuplicates(array));

* **Using IIFE:**

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

(function(arr) {

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

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

});

console.log(uniqueArray);

})(array);

* **Both versions will output an array with duplicates removed: [1, 2, 3, 4, 5].**

1. **Rotate an array by k times**

* **Using an anonymous function:**

const rotateArray = function(arr, k) {

const n = arr.length;

k = k % n;

const rotated = arr.slice(n - k).concat(arr.slice(0, n - k));

return rotated;

};

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

const k = 2;

console.log(rotateArray(array, k));

* **Using IIFE:**

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

const k = 2;

(function(arr, k) {

const n = arr.length;

k = k % n;

const rotated = arr.slice(n - k).concat(arr.slice(0, n - k));

console.log(rotated);

})(array, k);

* **Both versions will rotate the array by k times and output the rotated array: [4, 5, 1, 2, 3] for the provided array [1, 2, 3, 4, 5] and k = 2.**

1. **Do the below programs in arrow functions.**
2. **Print odd numbers in an array**

const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const printOddNumbers = arr => {

arr.forEach(num => {

if (num % 2 !== 0) {

console.log(num);

}

});

};

printOddNumbers(arr);

* **This program uses an arrow function printOddNumbers which takes an array as input and then iterates through each element using forEach method. Within the loop, it checks if the current number is odd or not using the modulus operator %. If the number is odd, it prints it to the console.**

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

const strings = ["hello", "world", "javascript", "programming"];

const convertToTitleCaps = strings => {

return strings.map(str => str.charAt(0).toUpperCase() + str.slice(1).toLowerCase());

};

console.log(convertToTitleCaps(strings));

* **This program defines an arrow function convertToTitleCaps which takes an array of strings as input. It uses the map method to iterate over each string in the array. Within the mapping function, it converts the first character of each string to uppercase using charAt(0).toUpperCase() and converts the rest of the string to lowercase using slice(1).toLowerCase(). Finally, it returns an array with all strings converted to title caps.**

1. **Sum of all numbers in an array**

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

const sumOfNumbers = numbers => {

return numbers.reduce((acc, curr) => acc + curr, 0);

};

console.log(sumOfNumbers(numbers));

* **This program defines an arrow function sumOfNumbers which takes an array of numbers as input. It uses the reduce method to iterate over each number in the array and accumulate their sum. The initial value of the accumulator (acc) is set to 0. Finally, it returns the sum of all numbers in the array.**

1. **Return all the prime numbers in an array**

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const isPrime = num => {

if (num <= 1) {

return false;

}

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) {

return false;

}

}

return true;

};

const primeNumbers = numbers.filter(num => isPrime(num));

console.log(primeNumbers);

* **This program defines an arrow function isPrime to check if a number is prime. Then, it uses the filter method to iterate over each number in the array and keeps only those that are prime. Finally, it prints the array containing all the prime numbers.**

1. **Return all the palindromes in an array**

const words = ["hello", "level", "world", "radar", "javascript"];

const isPalindrome = word => {

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

return word === reversedWord;

};

const palindromeWords = words.filter(word => isPalindrome(word));

console.log(palindromeWords);

* **This program defines an arrow function isPalindrome to check if a word is a palindrome. Then, it uses the filter method to iterate over each word in the array and keeps only those that are palindromes. Finally, it prints the array containing all the palindrome words.**