**DAY-4 ROAD-MAP TASK**

## Do the below programs in anonymous function , IIFE & arrow functions.

# Print odd numbers in an array

// // anonymous function

let oddNumbers = function(numbers){

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

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

console.log(numbers[i]);

}

}

}

let numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20];

oddNumbers(numbers);

output:

1

3

5

7

9

11

13

15

17

19

//-----------------------------------------------------------------------------------------------------------------------------------

// // IIFE with arrow function

((numbers) => {

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

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

console.log(numbers[i]);

}

}

})([1, 6, 3, 8, 5, 12, 7, 10, 9, 4]);

output:

1

3

5

7

9

//-----------------------------------------------------------------------------------------------------------------------------------

// // arrow function

let oddNumbers = (numbers) => {

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

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

console.log(numbers[i]);

}

}

}

let numbers = [1, 6, 3, 8, 5, 12, 7, 10, 9, 4, 15, 18, 21, 14, 17, 20, 23, 16, 25, 2, 19, 22, 11, 24];

oddNumbers(numbers)

output:

1

3

5

7

9

15

21

17

23

25

19

11

# Convert all the strings to title caps in a string array

// // anonymous function

let stringCaps = function(strings) {

let titleCapsArray = strings.map(strings => strings.charAt(0).toUpperCase() + strings.slice(1));

console.log(titleCapsArray)

}

let strings = ["apple", "banana", "orange", "grape", "kiwi"];

stringCaps(strings)

output:

[ 'Apple', 'Banana', 'Orange', 'Grape', 'Kiwi' ]

//------------------------------------------------------------------------------------------------------------------------------------------

// IIFE with anonymous function

(function(strings) {

let titleCapsArray = strings.map(strings => strings.charAt(0).toUpperCase() + strings.slice(1));

console.log(titleCapsArray);

})(["apple", "banana", "orange", "grape", "kiwi", "melon", "peach", "strawberry", "blueberry", "pineapple"]);

Output:

[

  'Apple',     'Banana',

  'Orange',    'Grape',

  'Kiwi',      'Melon',

  'Peach',     'Strawberry',

  'Blueberry', 'Pineapple'

]

//------------------------------------------------------------------------------------------------------------------------------------------

// // arrow function

let stringCaps = (strings) => {

let titleCapsArray = strings.map(strings => strings.charAt(0).toUpperCase() + strings.slice(1))

console.log(titleCapsArray)

}

let strings = ["hello", "world", "javascript", "programming", "openai", "chatbot", "array", "function", "algorithm", "variable"];

stringCaps(strings)

output:

[

  'Hello',      'World',

  'Javascript', 'Programming',

  'Openai',     'Chatbot',

  'Array',      'Function',

  'Algorithm',  'Variable'

]

# Sum of all numbers in an array

// // anonymous function

let sum = function (numbers){

total = 0;

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

total = total + numbers[i];

}

console.log(total);

}

let numbers = [2, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65];

sum(numbers);

output:

465

//-----------------------------------------------------------------------------------------------------------------------------------

// // IIFE with anonymous function

(function (numbers){

total = 0;

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

total = total + numbers[i];

}

console.log(total)

})([3, 8, 12, 17, 22, 27, 33, 40]);

Output:

162

//-----------------------------------------------------------------------------------------------------------------------------------

// // arrow

let sum = (numbers) => {

total = 0

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

total = total + numbers[i];

}

console.log(total)

}

let numbers = [2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53, 56, 59];

sum(numbers);

output:

610

# Return all the prime numbers in an array

// // anonymous function

let isPrimeNum = function (numbers) {

let findPrime = function (nums) {

if(nums <= 1 ) return false;

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

if(nums % i === 0) return false;

}

return true;

};

return numbers.filter(nums => findPrime(nums));

};

let numbers = [1, 6, 3, 8, 5, 12, 7, 10, 9, 4];

let primeNumbers = isPrimeNum(numbers);

console.log(primeNumbers);

output:

[ 3, 5, 7 ]

//-----------------------------------------------------------------------------------

// // IIFE

console.log((function (numbers) {

let findPrime = function (nums) {

if(nums <= 1 ) return false;

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

if(nums % i === 0) return false;

}

return true;

};

return numbers.filter(nums => findPrime(nums));

})([1, 6, 3, 4, 2, 17, 5, 11, 23, 7, 13, 3, 19, 29, 99, 12, 7, 232, 9, 52]))

Output:

[

   3, 2, 17, 5, 11,

  23, 7, 13, 3, 19,

  29, 7

]

//-----------------------------------------------------------------------------------

// Arrow Function

let isPrimeNum = (numbers) => {

let findPrime = function (nums) {

if(nums <= 1 ) return false;

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

if(nums % i === 0) return false;

}

return true;

};

return numbers.filter(nums => findPrime(nums));

}

let numbers = [1, 6, 3, 8, 5, 12, 7, 10, 7, 19, 2, 13, 31, 11, 5, 23, 17, 3, 9, 4];

let primeNumbers = isPrimeNum(numbers);

console.log(primeNumbers);

output:

[

   3,  5,  7, 7, 19,  2,

  13, 31, 11, 5, 23, 17,

   3

]

# Return all the palindromes in an array

// // anonymous function

let isPalindromes = function(word) {

let reverse = ''

for(let i = word.length - 1; i >= 0; i--){

reverse = reverse + word[i]

}

return word === reverse

}

let words = [

"level",

"hello",

"racecar",

"world",

"deified",

"apple",

"radar",

"banana",

"civic",

"computer"

];

let palindromes = [];

for (let word of words) {

if (isPalindromes(word)) {

palindromes.push(word);

}

}

console.log(palindromes)

output:

[ 'level', 'racecar', 'deified', 'radar', 'civic' ]

//-------------------------------------------------------------------------------------------

// // IIFE

console.log((function (words) {

let palindromes = [];

for (let word of words) {

let reverse = '';

for (let index = word.length - 1; index >= 0; index--) {

reverse = reverse + word[index];

}

if (word === reverse) {

palindromes.push(word);

}

}

return palindromes;

})([

"level",

"hello",

"racecar",

"world",

"deified",

"apple",

"radar",

"banana",

"civic",

"computer"

]));

Output:

[ 'level', 'racecar', 'deified', 'radar', 'civic' ]

//-------------------------------------------------------------------------------------------

// // Arrow Function

let isPalindromes = (word) => {

let reverse = '';

for(i = word.length - 1; i >= 0; i--){

reverse = reverse + word[i]

}

return reverse === word

}

let words = [

"level",

"hello",

"racecar",

"world",

"deified",

"apple",

"radar",

"banana",

"civic",

"computer"

];

let palindromes = [];

for(let word of words){

if(isPalindromes(word))

palindromes.push(word)

}

console.log(palindromes)

output:

[ 'level', 'racecar', 'deified', 'radar', 'civic' ]

## Do the below programs in anonymous function & IIFE

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

// // anonymous function

let isMediam = function(array1,array2){

let mergedArray = [...array1, ...array2].sort((a,b) => a - b);

length = mergedArray.length;

if(length % 2 === 0) {

let midIndex = length / 2 ;

return (mergedArray[midIndex - 1] + mergedArray[midIndex]) / 2

} else {

midIndex = Math.floor(length / 2)

return mergedArray[midIndex]

}

}

let array1 = [1, 3, 5, 7, 9];

let array2 = [2, 4, 6, 10, 8];

let results = isMediam(array1,array2);

console.log(results)

Output:

5.5

// ------------------------------------------------------------------------------

// // IIFE

console.log( (function(array1,array2){

let mergedArray = [...array1, ...array2].sort((a,b) => a - b);

length = mergedArray.length;

if(length % 2 === 0) {

let midIndex = length / 2 ;

return (mergedArray[midIndex - 1] + mergedArray[midIndex]) / 2

} else {

midIndex = Math.floor(length / 2)

return mergedArray[midIndex]

}

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

Output:

5.5

# Remove duplicates from an array

// // anonoymous Function

let removeDuplicateValue = function(duplicateValue) {

console.log([...new Set(duplicateValue)])

}

let duplicateValue = [3, 5, 2, 3, 7, 9, 5, 4, 2, 8];

removeDuplicateValue(duplicateValue)

output:

[

  3, 5, 2, 7,

  9, 4, 8

]

// ----------------------------------------------------------------------------------------------------

// // IIFE

(function(duplicateValue) {

console.log([...new Set(duplicateValue)])

})(["apple", "banana", "orange", "apple", "grape", "kiwi", "banana", "pear", "orange", "pineapple"]);

Output:

[ 'apple', 'banana', 'orange', 'grape', 'kiwi', 'pear', 'pineapple' ]

# Rotate an array by k times

// // anonymous Function

let arrayRotate = function(array, k) {

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

let firstValue = array[0];

for (let j = 0; j < array.length - 1; j++) {

array[j] = array[j + 1];

}

array[array.length - 1] = firstValue;

}

console.log (array);

}

let array = ['apple', 'banana', 'orange', 'grape', 'kiwi', 'pear', 'pineapple'];

let k = 2;

arrayRotate(array, k);

output:

[ 'orange', 'grape', 'kiwi', 'pear', 'pineapple', 'apple', 'banana' ]

//--------------------------------------------------------------------------------------------------------------

// // IIFE

(function(array, k) {

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

let firstValue = array[0];

for (let j = 0; j < array.length - 1; j++) {

array[j] = array[j + 1];

}

array[array.length - 1] = firstValue;

}

console.log (array);

})(['apple', 'banana', 'orange', 'grape', 'kiwi', 'pear', 'pineapple'],k = 2);

Output:

[ 'orange', 'grape', 'kiwi', 'pear', 'pineapple', 'apple', 'banana' ]

­­­