**ASSIGNMENT-14**

**Arrays in JavaScript:-**

**practice all discussed methods first and then make a doc on them.**

**Array:-**

Arrays in programming are data structures that store collections of elements. They provide a way to organize and manage related pieces of data under a single variable name. Arrays are widely used in programming because they offer flexibility, efficiency, and ease of access to their elements.

**Array Methods:-**

**1)splice() method:**

🡪Used to add new elements to array by deleting existing element.

🡪Modifies the existing array with remained elements

🡪contains more than two parameters

i)1st define index positions

ii)2nd Number of elements to be removed from given index

Example:=

let arr=[1,2,3,4,5,6]

arr.splice(2,1) //Output: [1,2,4,5,6]

arr.splice(2,3) //Output:[1,2,6]

**2)slice() method:**

🡪The slice() method in JavaScript is used to create a shallow copy of a portion of an array into a new array object selected from start to end (end not included) where start and end represent the index of items in that array.

🡪The original array will not be modified.

Example:=

let arr=[1,2,3,4,5,6]

let x=arr.slice(2,4)

console.log(x) //Output: [3,4]

let y=arr.slice(2)

console.log(y) //Output: [3,4,5,6]

**3)Array delete() method:**

🡪Using delete() leaves undefined holes in the array.

Example:=

let arr=[1,2,3,4]

delete arr[3]

console.log(arr) //Output:-[1,2,3,<empty item>,5]

🡪We don’t use delete() much,instead we use push and pop.

**4)Array flat() method:**

🡪Convert multi dimensional array to single dimentional array.

🡪Used for flatining of the array.

🡪flat returns new array by spplying flat.

Example:=

Let arr=[1,2,3,[4,5],[5,6],[6,7]]

let x=arr.flat(2)

console.log(x)

**Search Methods:-**

**1)Array indexof()**

**2)Array lastIndex()**

**3)Array includes()**

**1)Array indexof():-**

🡪gives index value of searching item

Example:=

let arr=[‘hi’, ‘js’, ‘css’, ‘react’]

let x = arr.indexof(‘js’) //1

let y=arr.indexof(‘html’) //-1

🡪if element not present in array it will return -1

**2)Array lastindexof():-**

🡪if elements are repeating it will find the last index of repeated elements.

Example:=

let arr=[1,2,2,3,3,1]

let x = arr.lastindexof(3)

console.log(x) //1

**3)Array includes():-**

🡪if searching element is present in array true else false

Example:=

let arr=[3,4,5,6,7,10]

let x = arr.includes(1)

console.log(x) //false

let y=arr.includes(4)

console.log(y) //true

**Map() Methods:-**

Map is a array iteration method, which includes call back function and executes the code for every iteration of element in the array and create a new array map() doesn’t modify original array.

Example:=

let arr=[1,2,3,4,5]

x=arr.map(function() {return ‘hi’})

console.log(x)

Output:

[‘hi’, ’hi’, ’hi’, ’hi’, ’hi’]

🡪Map method returns new Array.

🡪Map executes the length of the array for each iteration .

**1.Write a function squareNumbers(arr) that returns a new array where each number in the original array is squared using map method.**

let org\_arr=[2,3,4,5]

function squaredNumbers(org\_arr){

    return org\_arr.map(ele=>ele\*ele)

}

let new\_arr=squaredNumbers(org\_arr)

console.log(new\_arr)

**Output:**

[ 4, 9, 16, 25 ]

**2.Write a function toUppercase(arr) that returns a new array where each string in the original array is converted to uppercase using map method.**

let org\_arr=['html','css','javascript','python']

function toUpperCase(org\_arr){

    return org\_arr.map(ele=>ele.toUpperCase())

}

let new\_arr=toUpperCase(org\_arr)

console.log(new\_arr)

**Output:**

[ 'HTML', 'CSS', 'JAVASCRIPT', 'PYTHON' ]