**Arrays in JavaScript**

* Introduction

Adding new elements

Finding elements

Removing elements

Splitting elements

Combining elements

* Adding elements

const number = [3,4];

// adding element into array in last

number.push (5,6)

// adding element into array in start

number.unshift (1, 2);

// adding element into array in middle

number.splice (2 , 0 , 'a', 'b');

console.log(number);

* Finding elements (primitives)

  const number = [1,2,3,4,5];

    console.log(number.indexOf(6));

    console.log(number.indexOf(2));

    console.log(number.lastIndexOf(5));

console.log(number.indexOf(2) !== -1);  // old method

  console.log(number.includes(4));  // new method

* Finding elements (reference types)

  const courses = [

        {id : 1, name : 'a' },

        {id : 2, name : 'b' },

    ];

//   const course =  courses.find(function(course) {

//         return course.name === 'a';

// });

const course =  courses.findIndex(function(course) {

    return course.name === 'b';

});

 console.log(course);

* Arrow Function

    const courses = [

        {id : 1, name : 'a' },

        {id : 2, name : 'b' },

    ];

const course =  courses.findIndex (course => course.name === 'b');

 console.log(course);

* Removing elements

   const numbers = [1,2,3,4,5];

    //  remove element from end

    // const last =  numbers.pop();

    // remove element form start

    // const start = numbers.shift();

    // remove number in the middle

    numbers.splice (2,2);

    console.log(numbers);

* Emptying an Array

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

let another = numbers;

// solution 1

// numbers = [];

// solution 2

numbers.length = 0 ;              // solution 2 is most efficient

// solution 3

// numbers.splice (0 , numbers.length);

// solution 4

// while(numbers.length > 0)

//     numbers.pop();

//

console.log(numbers);

console.log(another);

* Combining and slicing Arrays

// const first = [1,2,3];

const first = [{id:1}];

const second = [4,5,6];

// combining an array

   const combine =  first.concat(second);

   first[0].id = 10;

//  slicing an array

//   const slice = combine.slice(2,4);  // first way

//   const slice = combine.slice(3);    // second way

     const slice = combine.slice();   // third way

  console.log(combine);

  console.log(slice);

* The spread Operator

    // spread operator is more efficient than concat method

    const first  = [1,2,3];

    const second = [4,5,6];

    const combined = [...first , 'a', ...second , 'b'];

    const copy = [...combined];

    console.log(combined);

    console.log(copy);