**6. Arrays**

**1) Adding Element**:

We can add element in an array at beginning, middle or at the end position. For this purpose we have three methods.

1. push() => Add element at the end
2. unshift() => Add element at the beginning
3. splice() => remove, replace or add element on middle.

**Example**:

const number = [3, 4];

//End-> add element at the end

number.push(5, 6);

console.log(number); //[ 3, 4, 5, 6 ]

//Beginning-> add element at the beginning

number.unshift(1, 2);

console.log(number); //[ 1, 2, 3, 4, 5, 6 ]

//Middle-> add element in middle

number.splice(2, 0, 3.5);

console.log(number); [ 1, 2, 3.5, 3, 4, 5, 6 ]

**splice() method**:

The splice() method changes the contents of an array by removing or replacing existing elements and/or adding new elements.

splice(start: number, deleteCount: number, ...items: number[])

start -> from which index we want to start

deleteCount -> how many element we want to delete

items -> element we want to add

**2) Finding element (Primitive Type)**:

Finding element really depends what we are works with primitive type or reference type in an array. In primitive type for find element in an array we can use the following method.

**a) indexOf()**:

The indexOf() method searches the array for the specified item, and returns its position (index).

The search will start at the specified position, or at the beginning if no start position is specified, and end the search at the end of the array.

Returns -1 if the item is not found.

If the item is present more than once, the indexOf method returns the position of the first occurrence.

Note: The first item has position 0, the second item has position 1, and so on.

**Example**:

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

//find the index of element 3 => 1

console.log(number.indexOf(3)); //1

//return -1 if element not present in array

console.log(number.indexOf(11)); //-1

//start find from index 2 and find index of element 7 => 5

console.log(number.indexOf(7, 2)); //5

//if more than one element return the index of first element

console.log(number.indexOf(2)); //0

**b) lastIndexOf()**:

The lastIndexOf() method searches the array for the specified item, and returns its position.

The search will start at the specified position or at the end if no start position is specified, and end the search at the beginning of the array.

Returns -1 if the item is not found.

If the item to search for is present more than once, the lastIndexOf method returns the position of the last occurrence.

6. Arrays