Arrays

--------

> array is collection of elements (values).

> storing group of values with same refname is called array.

> array allows similar type of values (homogeneous) as well as different types of values, means one array can store group of numbers, strings, booleans etc...

> we can arrays create in local scope or outer scope.

> arrays are belongs to reference/non-primitive datatype.

> primitive dt stores data but non-primitive stores address of data.

> arrays are created dynamically, and arrays are created in heap area.

> arrays maintain data in sequence order

adv:

> arrays are simplyfying coding when work with group of values.

> easy transporting data

> also used for data maintenance in applications

array creation:

Approach 1:

using array Literals [ ]

Syn: let/var/const array = [ ];

let/var/const array = [val1,val2,val3, ...];

Approach 2:

using new kw & constructor

Syn:- var/let/const array = new Array();

var/let/const array = new Array(val1,val2,...);

accessing array:

array[index]

index is a slno of memory block, its start 0.

set value:

array[index]=value;

size of array:

array.length ==> predefined property, it returns size of array

Associative Arrays:

If you use a named index when accessing an array, JavaScript will redefine the array to a standard object, and some array methods and properties will produce undefined or incorrect results.

MDA

--------

storing group of ele in tabler (row & col) format is called MDA (2DA).

mda is a coll of sda's

array creation:

var array=[ [val1, val2, ...],

[val1, val2, ...],

...

];

accessing array:

array[rowind][colind]

set value:

array[rowind][colind]=value;

size of array:

array.length => it returns no.of rows

array[rowind].length => it returns no.of cols

array methods

--------------------------

pop()

it returns ele of array (R -> L), it removes popped ele

array.pop()

shift()

it returns ele of array (L -> R), it removes shifted ele

array.shift();

unshift()

add a new element @begining of array

array.unshift(value);

indexOf()

finding given ele ava in an array or not

if found => index, 1st occurence

if not found => -1

by def search starts from 0th index or search starts from given index.

lastIndexOf()

finding given ele ava in an array or not

if found => index, last occurence

if not found => -1

include()

it searching the given ele found or not

if found => true

not found => false

sort()

it sorting an array in asce order

reverse()

it re-arrange ele of array in reverse order

splice()

it used to remove/delete ele from an array based given index

array.splice(st-index, no.of elements)

it used to insert ele in array based given index

array.splice(index, 0, newvalue)

it used to overwrite eles of array

join()

this method creates and returns a new string by concatenating all of the elements in an array (or an array-like object), separated by commas or a specified separator string. If the array has only one item, then that item will be returned without using the separator.