Arrays

> array is collection of elements (values).

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

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

adv:
- > arrays are simplifying coding when work with group of values.
- > easy transporting data
- > also used for data maintenance in application

> arrays we can create local scope or outer scope.

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

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

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

Syn:
array creation:
Approach1 (using Literals [ ]):
```
let/var/const  array = [ ];
let/var/const  array = [val1, val2, val3, ...];
```

Approach2 (using new kw):
```
var array = new Array();
var array = new Array(val1,val2,...);
```

```
datatype array[size];    <== c/c++
      datatype array[] = new datatype[size];  <== java
```

accessing array:
```
array[index]
index is a slno of memory block, its start 0.
```
set value:
```
array[index]=value;
```
size of array:
```
array.length          ==> predefine property, it returns size of array
array.length=N;   ==> it reset size of array
```

push()
  add a new element @end of array
```
array.push(newvalue)
```

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()




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


for in loop
----------
> it used to get elements from an array based on index
> this loop extracting elements in forward direction only (back not sup)
> we can't start the loop from middle of array (random access not poss)
Syn:
```

```
      for(var in array)
      {
        code
      }
```

for of loop

> it used to get elements from an array based on value
> this loop extracting elements in forward direction only (back not sup)
> we can't start the loop from middle of array (random access not poss)

Syn:

```
      for(var of array)
      {
        code
      }
```

forEach loop

> this loop used to get elements from an array based on value
> this loop extracting elements in forward direction only (back not sup)
> we can't start the loop from middle of array (random access not poss)

Syn:

```
      array.forEach(function(variable){
                  code
               });
```