

14/5/21

Page No.

Date:

## \* Data-Structures...

- Majorly they do the work of storing data!
- And we do so, in order to make sure that our operations perform efficiently.
- Every data structure has it's own way of storing data.
- In Javascript we have "Collections".
- In "Collections" we are already provided with the data structures which are implemented.

- ① Arrays
- ② Stack
- ③ Queue
- ④ Maps
- ⑤ Set

## \* Arrays "Collections In JavaScript"

- ① Indexed Collections → We access the elements using indexes.
- ② Keyed Collections. → In here we access the elements using keys.

~~Ex~~

Indexed → Arrays

Keyed → Maps, Sets.

(1) Indexed → Arrays.

→ We have marks of students which need to be stored.

0	1	2	3	4
10	20	30	40	90

So, to get 20 we can use index '1'.

(2) Keyed → Maps, Set.

→ We have to store name of the students along with their roll no.

Ex → "Rutvi" 19 <sup>→ value</sup>, "Mayank" 1  
Key 1 "Komal" 18

"To access values we must use keys."

## \* Arrays →

- An array is an ordered list of values that you refer to with a name and an index.
- An array is an ordered list of values stored in continuous memory location.  
OR

Indexes →	0	1	2	3	4	5	6	7	8	9
arr →	11	34	25	56	78	89	90	54	67	78

- If your array is having '12' elements then the last element will be stored at index '11' as indexing in arrays start from '0'.

## \* To access 67 we can do is →

arr[8]  
↳ index no.  
name of arr

## How to declare an Array →

- Three ways →
- ① let arr = new Array(element0, element1, ..., elementN)
  - ② let arr = Array(element0, element1, ..., elementN)
  - ③  let arr = [element0, element1, ..., elementN]  
most used

\* When we know the length of the array  
and not elements →

- ① let arr = new Array(arrayLength)
- ② let arr = Array(arrayLength)
- ③ let arr = [ ]

arr.length = 42;

arr.length = arrayLength.

Note → We can store multiple data types  
in an array in javascript.

Ex →

~~name~~ names = [26, "Rutvi"];