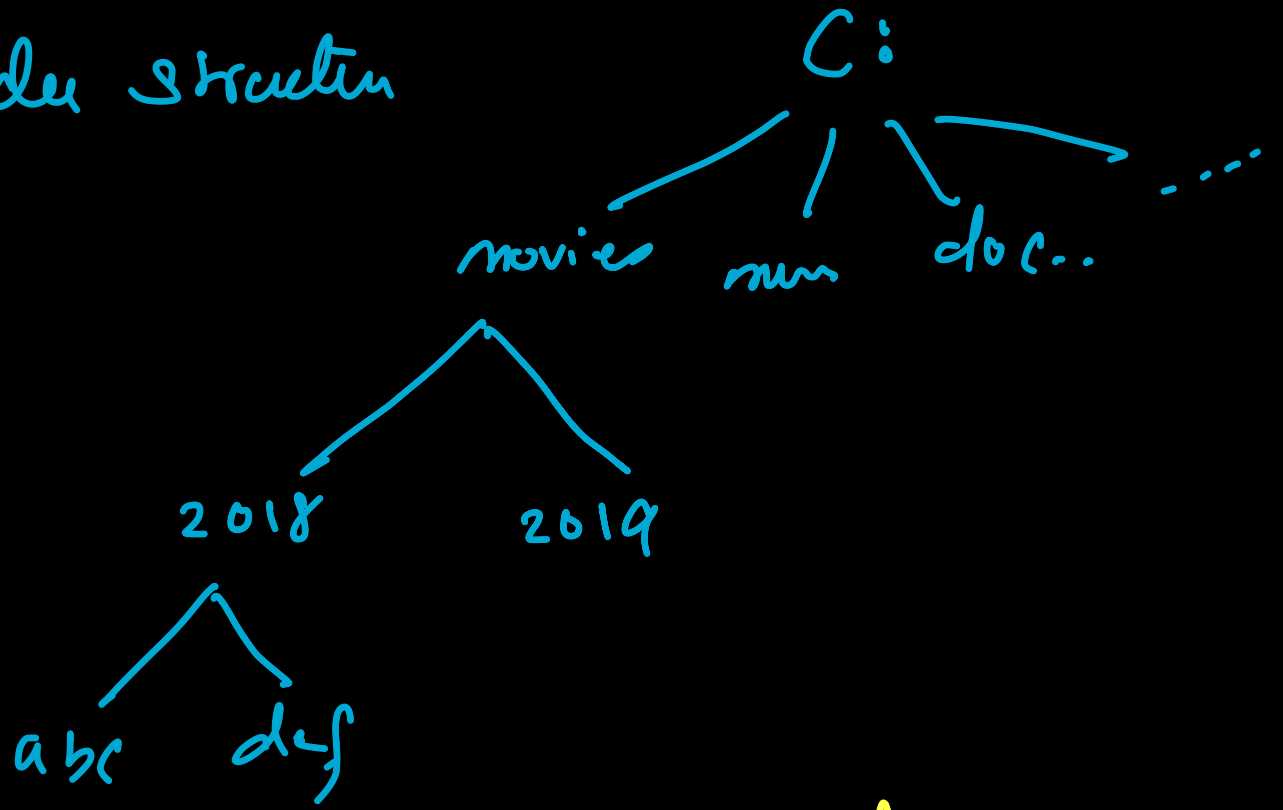
Arrays

Data Structures → It represents a mental model to store & retreive data in a particular way.

Hierarchy based data : → folder structure

↳ Trees

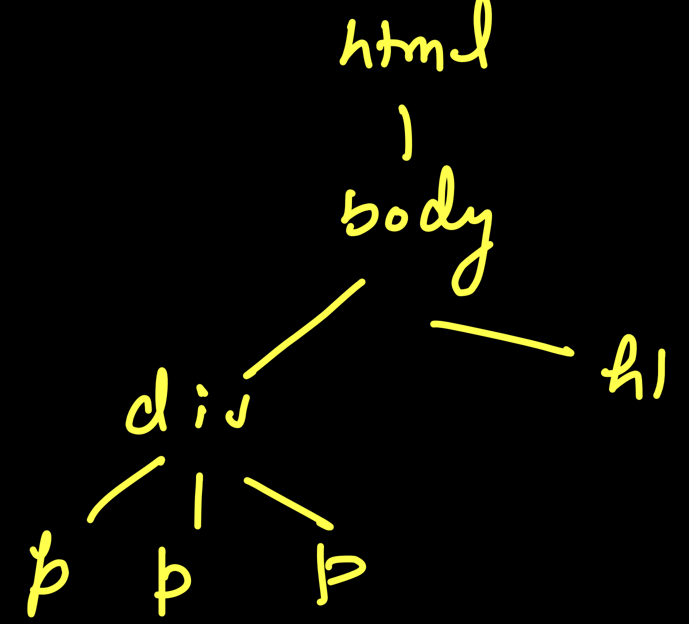


HTML DOM

↳ load → Browser

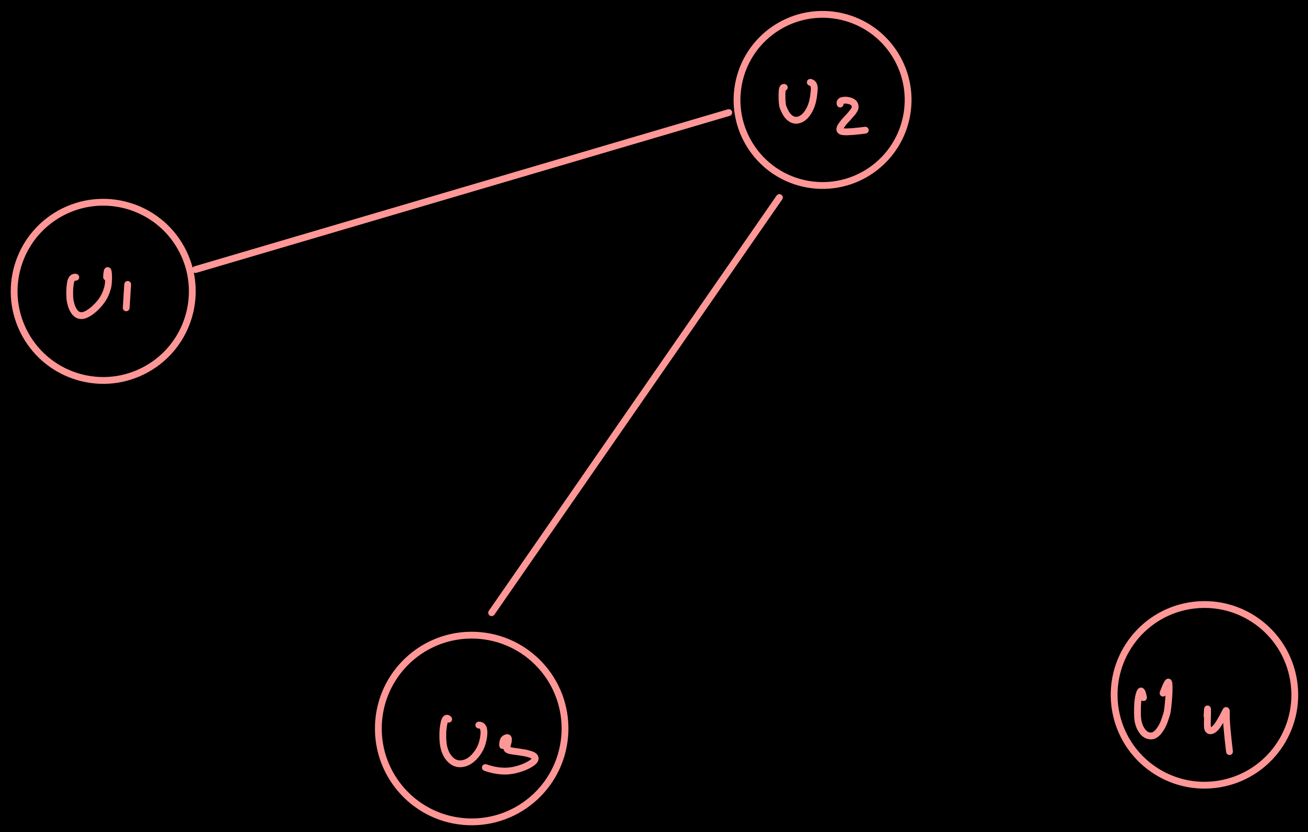
↓ parsed in the form of a tree

```
<html>
  <body>
    <div>
       <p>
       <p>
       <p>
    </div>
    <h1> </h1>
  </body>
</html>
```



graph representation:

Graph



Arrays

↳ Linear data structure (1D space)

↳ In some languages like C++, Java they are homogenous

only one type of data is stored

& in some other like JS, Python they are heterogenous

diff type of data can be stored

["abc", ~~1~~ 10, null, 2, false...]

↳ Array are mutable means data inside array can be updated.

↳ Arrays are used to implement a lot of other data structures

994

Ex → Heaps (PQ)

- ↓ mental model ↓ hierarchy
- → arrays

Segment Tree

- ↓ mental model ↓ hierarchy
- → arrays

Stack

- → arrays

↳ arrays can be easily manipulated to make it multidimensional;

Images

Image

arrays

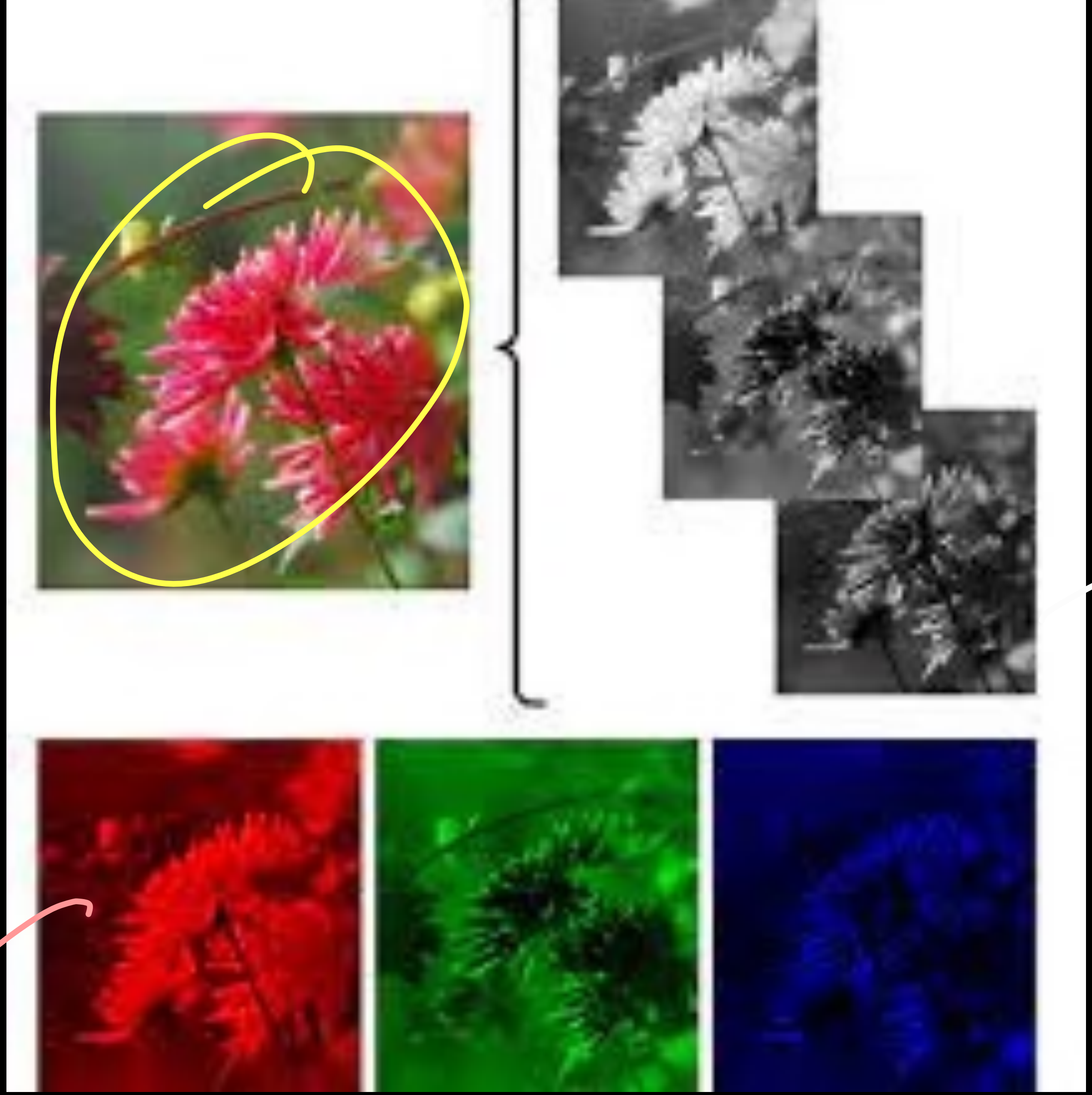
Matrix

2D array

Intensity

0-255

0		33	244	1	3		



Red

green

Blue

ML

2d arrays of pixel values

3d

3 layers

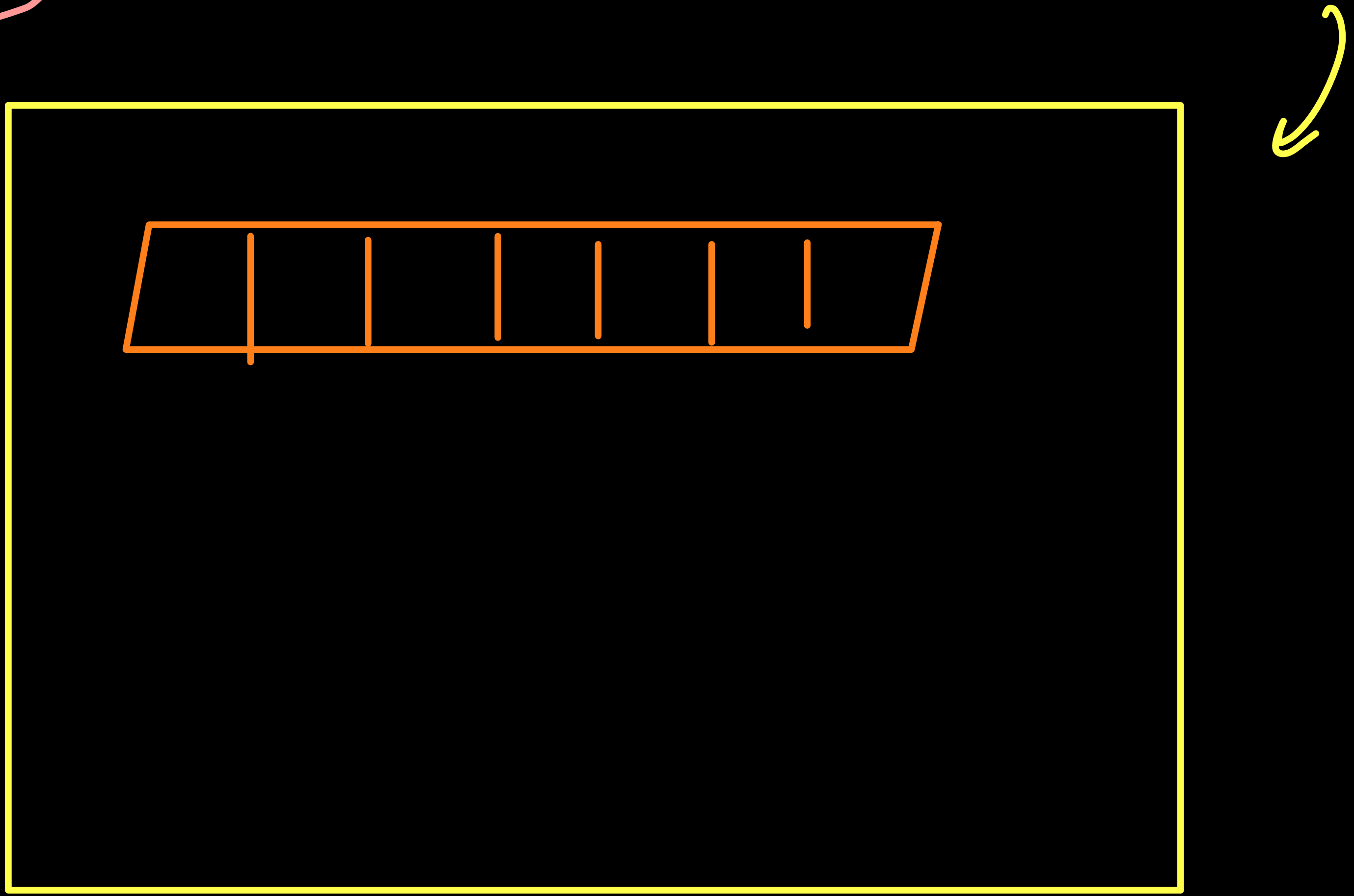
Satellite Images

SIH 2019

→ ISRO

→ Arrays are linear, thus in memory (RAM) arrays consume contiguous memory loc.

continuous



Array in DS follows indexing to identify each element.

2

[0, false, 3, "Samrat", null] ← elements

0 1 2 3 4 ← indexes / positions

0-based indexing } → counting starts with 0

Using indexes we can fetch value present at that index.

```
arr = [3, 4, 9, 1]

arr.forEach((element, idx) => {
        //some ops
        console.log("Element is", element);

})
```

optional index arg

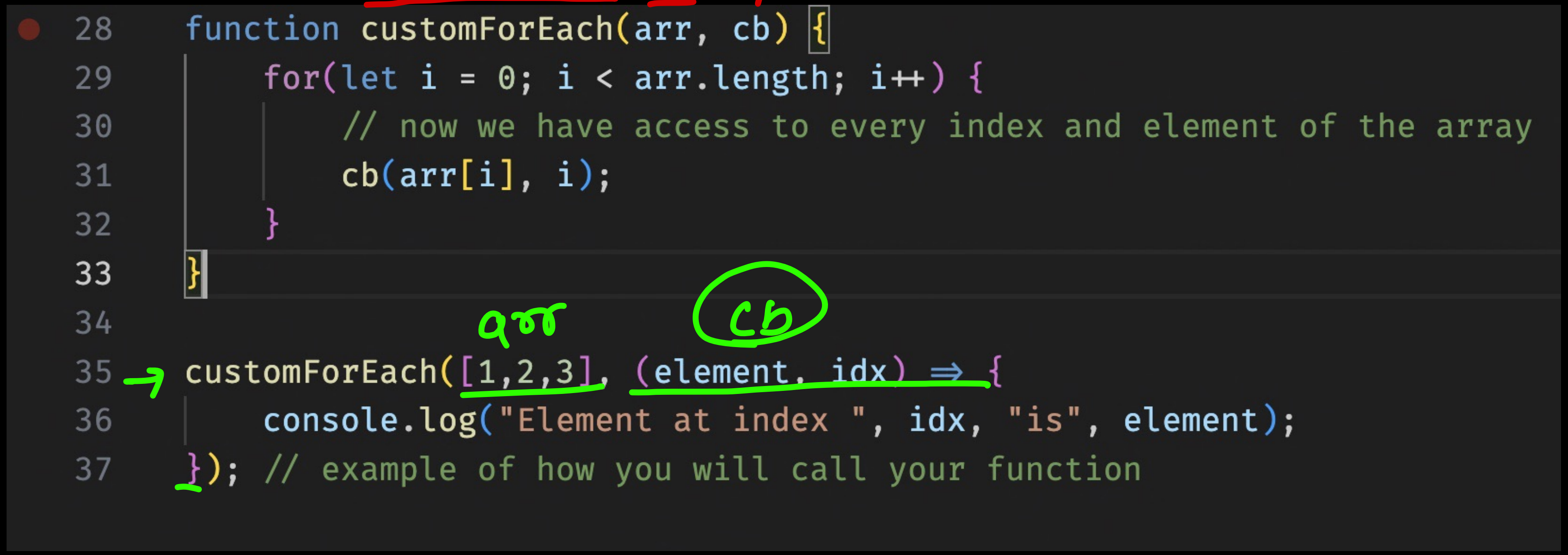
goes to every element of the given array. On that elem calls the callback.

```
Element is 3
Element is 4
```

HoF

array

callback



```
function customForEach(arr, cb) {
    for(let i = 0; i < arr.length; i++) {
        // now we have access to every index and element of the array
        cb(arr[i], i);
    }
}

customForEach([1,2,3], (element, idx) => {
    console.log("Element at index ", idx, "is", element);
}); // example of how you will call your function
```

arr

cb

[1, 2, 3]

i = ~~0~~ 1

(1, 0)

(2, 1)