Arrays

Q: What are arrays ??

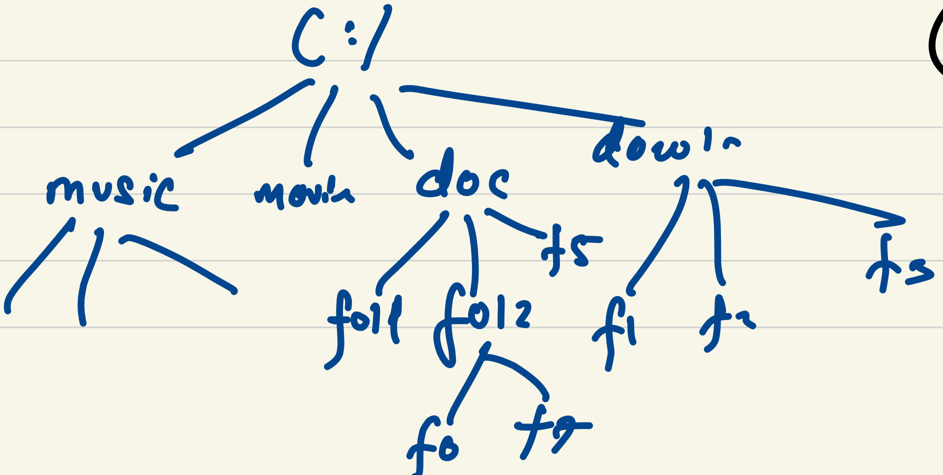
→ Let's say we have to store 100 numbers.

analysing stock price

→ Arrays can be described as a data structure that can store collection of data, in a linear contiguous orientation in the memory.

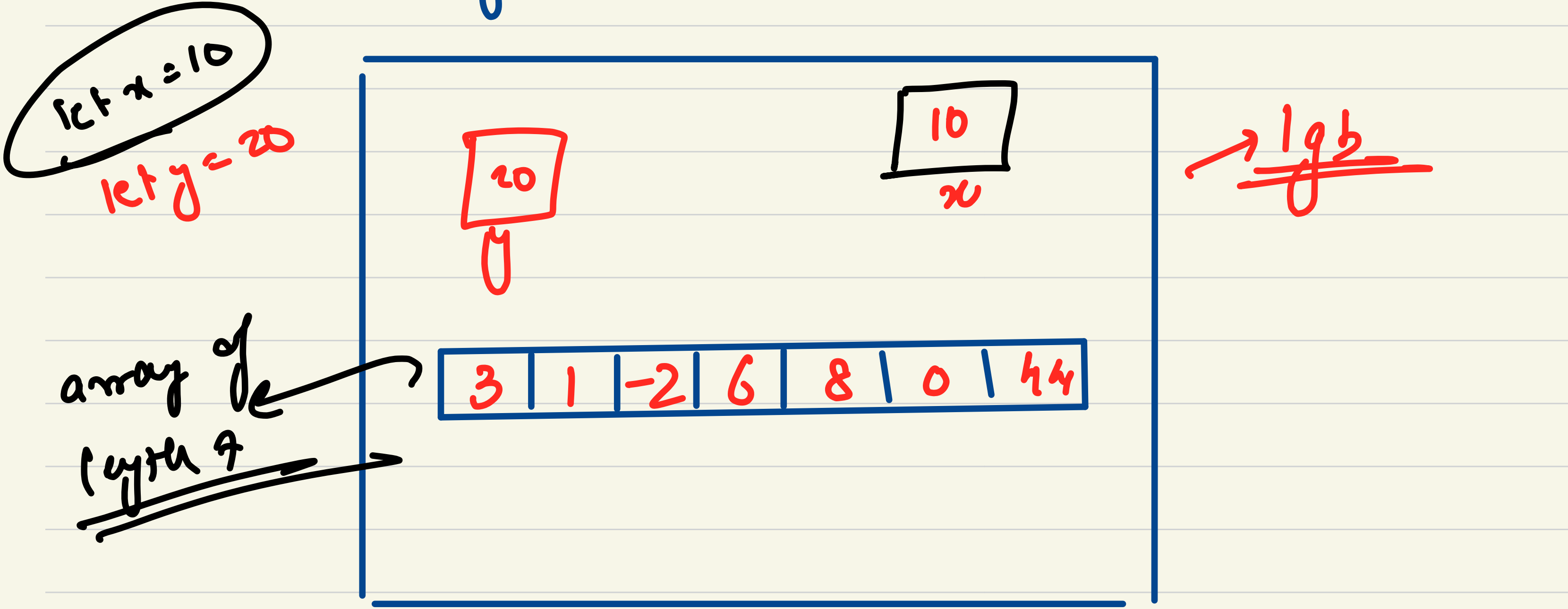
# data structure → These are different structures mind map that we can prepare to store data in different ways depending on our use case.

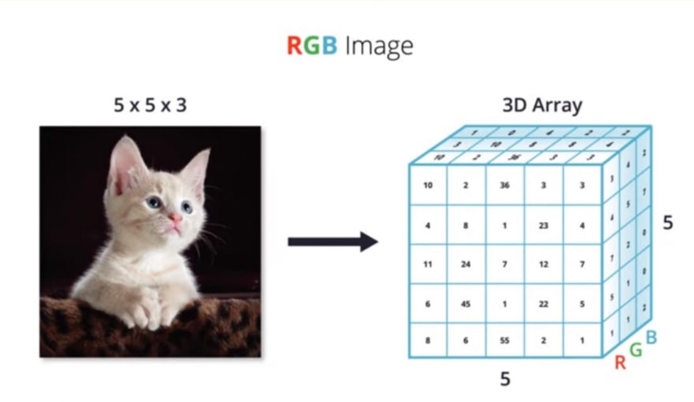
Or → **Hierarchy** → folder structure



Trees

Array → Arrays actually store data in a linear contiguous orientation in the memory.



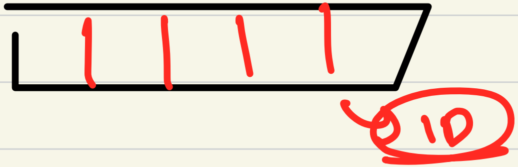


RGB Image

5 x 5 x 3

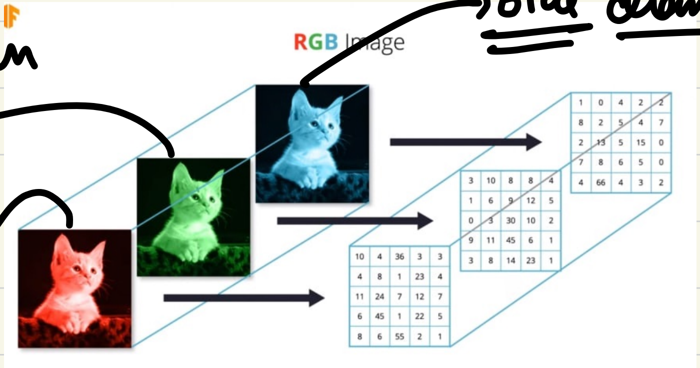
3D Array

10	2	36	3	3
4	8	1	23	4
11	24	7	12	7
6	45	1	22	5
8	6	55	2	1



blue channel

green



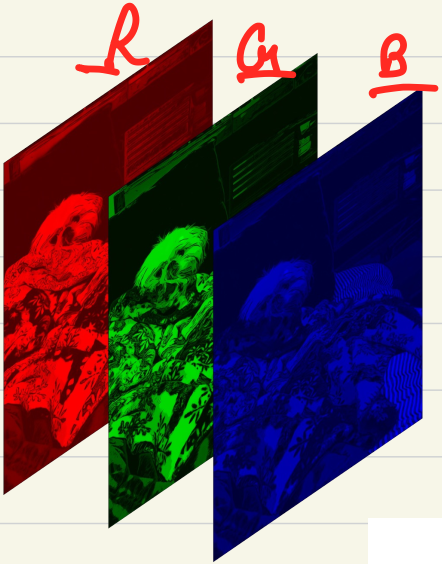
RGB Image

10	4	36	3	3
4	8	1	23	4
11	24	7	12	7
6	45	1	22	5
8	6	55	2	1

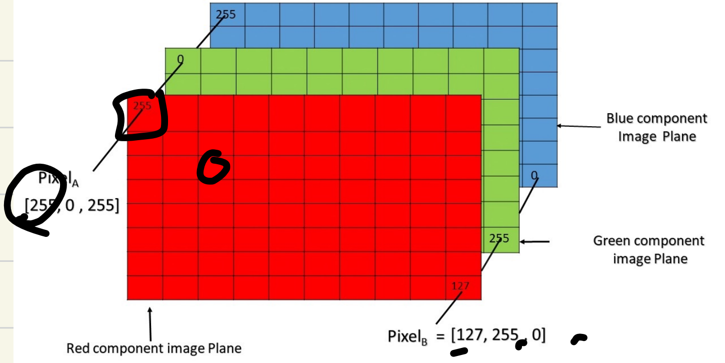
3	10	8	8	4
1	6	9	12	5
0	3	30	10	2
9	11	45	6	1
3	8	14	23	1

1	0	4	2	2
8	2	5	4	7
2	13	5	15	0
7	8	6	5	0
4	66	4	3	2

Red channel



$\boxed{0,255}$


255
0
255
Pixel$_A$
[255, 0 , 255]
Blue component Image Plane
0
255
Green component image Plane
127
Red component image Plane
Pixel$_B$ = [127, 255, 0]

Pixel of an RGB image are formed from the corresponding pixel of the three component images

How do we create arrays in JS??

```
let x = [ ];
```

empty array of 0 length

```
var y = [1, 2, 4, 6, false, "abc", true, null];
```

8 length array

```
const z = ["ab", "cd", "dc"]
```

array of length 3

Note → JS arrays support heterogenous values

Indexing

arr → [1, 2, 3, 4, 5, 6]

index → 0 1 2 3 4 5

arr[4] → 5

individual elements of the array has got unique indexes. Starting from 0.

array = ["abc" , 2, 5, 18, undefined , 0]

index → 0 1 2 3 4 5

array[3] → 18

0 1 2 3 4 5

str = "Sanket"

arr = [0, -1, 3, 8, 2]

0 1 2 3 4

access data → strings → str[2] → "n"

arrays → arr[2] → 3

arr[4] → 2

update the data → strings → str[0] = "b" ✗ not possible

arr[2] = 11 → ✓ possible

updation is possible inside arrays but not strings.

string in JS are immutable

arrays in JS are mutable.

Qn Given an array which only contains 0 & 1.

The data is shuffled randomly.

Write a function that can rearrange the data

Such that all the 0's are present before 1.

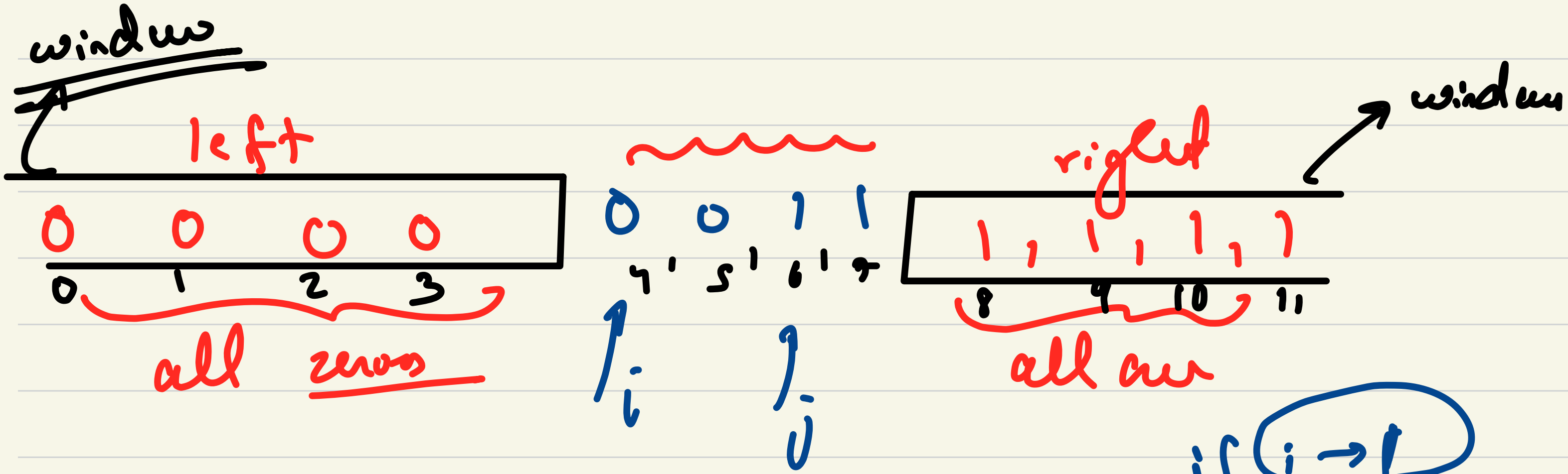
Ex → arr → [1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1] (don't create new array)

↓

arr → [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]

Condition → Do the rearrangement, by reading / accessing the whole array only once! (Single pass)

[1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1]

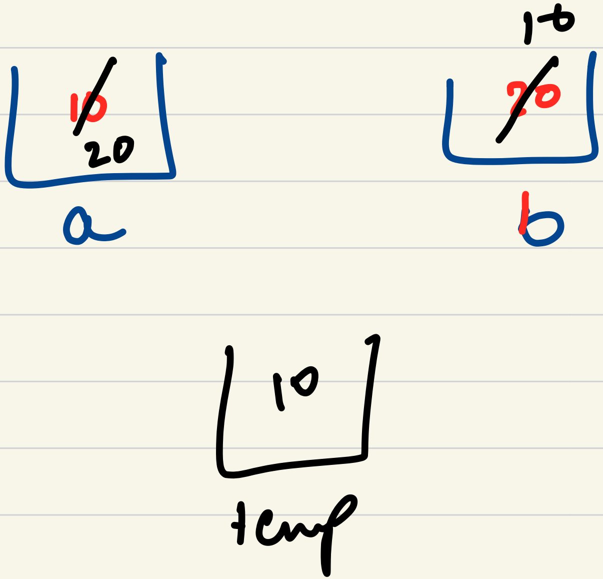


```
if (i → 1)
swap → value → i, j
if (arr[i] == 1)
    swap
    j--;
if (arr[i] == 0)
    i++
```

i, j → represent indexes

befor i, every thing is a zero

after j, every thing is a one.



temp = a
a = b
b = temp

↳ # for..of **loop**

```
arr = [10, 11, 15, 20];

  for (const value of arr) {
        console.log(value)
  }
```

Objects → (key value)

```
let x = { name: "Sanket" , age: 24 }

let y = {
    name : "Sarthak",
    age : 24
};
```

put key as a string, it return the value

```
x["name"] → "Sanket"

x.name → "Sanket"
```

How to access key value pairs??

why we cannot access a value from a key ??

↳ in an object key are unique, but multiple diff keys can have same value in a object.

→ Given a string, try to print no. of occurences of each unique character of a string.

Ex→ javascript

ans →

j	1
a	2
v	1
s	1
c	1
r	1
i	1
p	1
t	1

//

printing can be done in any order.

How about we somehow store frequency of each unique char. We need a mapping based structure.

< key → value >   Objects

frequencyMap

javascript

↑
i

{
'j' : 1,
'a' : ~~1~~ 2, → incremented the value.
'v' : 1,
'S' : 1,
}

```
let freqMap = { };                     → initially empty
for ( const  char  of  str ) {
    if (freqMap.char ) {               → if char is not prent we
                                         have a false
        freqMap.char +=1 ;             → update

    } else {
        freqMap.char =1 ;              → add a new
                                         key,value
    }
}
```